Classical Predicative Logic-Enriched Type Theories[☆]

Robin Adams[*], Zhaohui Luo

Royal Holloway, University of London

Abstract

A logic-enriched type theory (LTT) is a type theory extended with a primitive mechanism for forming and proving propositions. We construct two LTTs, named LTT_0 and LTT_0^*, which we claim correspond closely to the classical predicative systems of second order arithmetic ACA_0 and ACA. We justify this claim by translating each second-order system into the corresponding LTT, and proving that these translations are conservative. This is part of an ongoing research project to investigate how LTTs may be used to formalise different approaches to the foundations of mathematics.

The two LTTs we construct are subsystems of the logic-enriched type theory LTT_W, which is intended to formalise the classical predicative foundation presented by Herman Weyl in his monograph *Das Kontinuum*. The system ACA_0 has also been claimed to correspond to Weyl's foundation. By casting ACA_0 and ACA as LTTs, we are able to compare them with LTT_W. It is a consequence of the work in this paper that LTT_W is strictly stronger than ACA_0.

The conservativity proof makes use of a novel technique for proving one LTT conservative over another, involving defining an interpretation of the stronger system out of the expressions of the weaker. This technique should be applicable in a wide variety of different cases outside the present work.

Keywords: type theory, logic-enriched type theory, predicativism, Hermann Weyl, second order arithmetic
2000 MSC: 03B15, 03B30, 03B70, 03F25, 03F35, 68T15

1. Introduction

A lot of research in the field of mathematical logic has been devoted to constructing formal theories intended to capture various schools of thought in the

[☆] This research was supported by the UK EPSRC research grant EP/D066638/1 and research grant F/07-537/AA of the Leverhulme Trust in the UK.

[*] Corresponding author. Address: Department of Computer Science, Royal Holloway, University of London, Egham Hill, Egham, Surrey, TW20 0EX, United Kingdom. Tel: +44 1784 443421. Fax: +44 1784 439786

Email addresses: robin@cs.rhul.ac.uk (Robin Adams), zhaohui@cs.rhul.ac.uk (Zhaohui Luo)

foundations of mathematics. In particular, the project of Reverse Mathematics [1] has provided an extremely detailed analysis of many theories in the language of *second-order arithmetic* Ł. It has been argued that the theories studied correspond closely to different foundational schools; in particular, that the classical, predicative foundation presented by Hermann Weyl in his monograph *Das Kontinuum* [2] is captured by the theory ACA_0 [3].

The systems of logic known as *dependent type theories* have also received a lot of attention, and in particular have proven to offer many practical benefits when used as the basis of the computer systems known as *proof checkers* or *proof assistants*. Type theories divide the world of mathematical objects into *types*. They offer much more expressive power than second-order arithmetic: we are able to speak, not just of natural numbers and sets of natural numbers, but also about (e.g.) sets of sets, lists, trees, and functions from any of these types to any of them. However, so far, type theories have been used almost exclusively to represent *constructive* mathematics.

More recently, the concept of a *logic-enriched type theory* has been developed. A logic-enriched type theory is a type theory augmented with a separate, primitive mechanism for forming and proving propositions. It thus has two components or 'worlds': a type-theoretic component, consisting of objects collected into types, and a logical component, for reasoning about these objects. LTTs have been used to investigate the relationships between type theories and set theories [4, 5], and by the present authors [6, 7] to formalise the predicative foundation for mathematics presented by Hermann Weyl in *Das Kontinuum* [2].

There is reason to believe that LTTs may offer some of the advantages of both traditional logical systems, and type theories. They share with type theories the rich type structure and inbuilt notion of computation that have proven to be of great benefit for formalisation in practice. At the same time, they offer the flexibility in choice of axioms that we are used to in traditional logical systems: it is possible, for example, to add excluded middle to the logical component without changing the type-theoretic component.

This paper is part of an ongoing research project to construct a hierarchy of LTTs, similar to the hierarchy of second-order systems in Reverse Mathematics. We hope thereby to investigate how LTTs may be used to represent different schools of thought in the foundations of mathematics, and to understand the effect that changes in the design of an LTT have on its set of definable objects and provable theorems.

In this paper, we construct two LTTs that capture two second-order systems that are closely related to the foundation of *Das Kontinuum*: ACA_0 and ACA. We construct two LTTs, which we name LTT_0 and LTT_0^*. These are more expressive than a second-order system: the type-theoretic component of each features types of natural numbers, pairs, functions of all orders, and sets of all orders.

Our aim in this paper is to show that adding this expressive power is 'safe'; that is, that we have not thereby increased the proof-theoretic strength of the system. We do this as follows. Let us say that a proposition of LTT_0 is *second-order* iff it uses no types other than $\mathbb{N}$ (the type of natural numbers) and $\mathrm{Set}\,(\mathbb{N})$

(the type of sets of natural numbers). We define a translation from ACA_0 onto the second-order propositions of LTT_0, and prove that the translation is *conservative*; that is, a formula of Ł is provable in ACA_0 if and only if its translation is provable in LTT_0.

The current authors have previously [6, 7] presented a new system intended to capture Weyl's foundation, which we named LTT_W. We argued there that LTT_W captures Weyl's foundation very closely, and described how all the definitions and results in *Das Kontinuum* have been formalised in LTT_W using a proof assistant. The two LTTs that we construct in this paper are both subsystems of LTT_W. As a consequence of the work in this paper, we now know that LTT_W is strictly stronger than ACA_0, and at least as strong as ACA.

We argue that, compared with ACA_0 and ACA, LTT_W corresponds more closely to the system presented in *Das Kontinuum*. This is not a claim that can be proven formally, as there is no formal definition of Weyl's foundation, but we can advance evidence for it. In our previous paper, we pointed out the extreme similarity between the presentation in *Das Kontinuum* and the definition of LTT_W, and described one construction in *Das Kontinuum* — the construction of $K(n) = \{X \mid X$ has at least n elements$\}$ — that cannot be done 'as directly' in any of the second order systems. Here, we strengthen the justification for this claim: we show that K is expressed by a term in LTT_W that cannot be formed in either LTT_0 or LTT_0^*.

The majority of this paper is taken up with proving the conservativity results. Our method for proving the conservativity of LTT_0 over ACA_0 is as follows. We first define a subsystem T_2 of LTT_0 which has just two types, $\mathbb{N}$ and $\mathrm{Set}\,(\mathbb{N})$. and show that LTT_0 is conservative over T_2.

We then construct infinitely many subsystems of LTT_0 between T_2 and LTT_0. We prove that, for each of these subsystems S and T, whenever S is a subsystem of T, then T is conservative over S. We do this by defining an interpretation of the judgements of T in terms of the expressions of S. Informally, we can think of this as giving a way of reading the judgements of T as *statements about* S. We show that this interpretation satisfies two properties:

- Every derivable judgement of T is true.
- Every judgement of S that is true is derivable in S.

It follows that, if a judgement of S is derivable in T, then it is derivable in S.

The proof thus makes use of an original technique which should be of interest in its own right, and which we expect to be applicable in a wide variety of contexts for proving one LTT or type theory conservative over another. In particular, we shall show how it can be adapted to provide a direct proof that ACA_0 is conservative over Peano Arithmetic.

1.1. Outline

In Section 2 of this paper, we describe the subsystems of second order arithmetic that we shall consider, and compare them informally with Weyl's system. In Section 3, we give the formal definition of LTT_W and its two subsystems, and define the translation from second order arithmetic into the LTTs. In Section 4, we prove that this translation is conservative in the case of ACA_0 and T_2. In Section 5, we prove that LTT_0 is conservative over T_2. Finally, in Section 6, we indicate how the proof can be modified to prove the conservativity of LTT_0^* over ACA, and discuss the possibility of constructing a subsystem of LTT_W conservative over ACA_0^+, and the conservativity of ACA_0 over Peano Arithmetic.

Notation. We shall stick to the following convention throughout this paper. Capital letters from the beginning of the Latin alphabet (A, B, C, $\ldots$) shall denote types. Capital letters from the middle (K, L, M, N, $\ldots$) shall denote terms. Capital letters from just after the middle (P, Q) shall denote names of small propositions. Lower-case letters (x, y, z, $\ldots$) shall denote variables, except t, which we reserve for terms of the language of second-order arithmetic. Lower-case letters from the middle of the Greek alphabet (ϕ, ψ, χ, $\ldots$) shall denote propositions.

We shall be dealing with partial functions throughout this paper. We write $X \simeq Y$ to denote that the expression X is defined if and only if Y is defined, in which case they are equal. Given a function v, we write $v[x := a]$ for the function v' with domain $\mathrm{dom}\, v \cup \{x\}$, such that $v'(x) = a$, and $v'(y) = v(y)$ for $y \neq x$. We write $\mathrm{FV}\,(X)$ for the set of free variables in the expression X.

2. Background

2.1. Weyl's Das Kontinuum

In 1918, Herman Weyl wrote the monograph *Das Kontinuum* [2], which presented a semi-formal system intended to provide a predicative foundation for mathematics. Weyl's system consists of a set of 'principles' by which sets, functions and propositions may be introduced. In particular, if we have formed the proposition ϕ, we may introduce the set $\{x \mid \phi\}$, provided that ϕ does not involve any quantification over sets. Impredicative definitions are thus impossible in Weyl's system. His concern was to show how much of mathematics — in particular, how much of analysis — could still be retained under such a restriction.

At the time of writing *Das Kontinuum* in 1918, Weyl agreed with Whitehead and Russell's opinion [8] that the source of the famous paradoxes in set theory was the presence of *impredicative definitions* — definitions that involved a certain kind of *vicious circle*. In particular, when we introduce a set R with the definition

$$R = \{x \mid \phi\} \tag{1}$$

then the definition is impredicative if either x or any of the bound variables in ϕ ranges over a collection that includes the set R itself.

In Weyl's foundation, mathematical objects are divided into *categories*. A category can be *basic* or *ideal*. Given any category A, there is the ideal category

Set (A)[1] of *sets* whose members are objects of category A. In a definition of the form (1), we may only quantify over *basic* categories. In particular, we may not quantify over any category of the form Set (A). It is in this manner that impredicative definitions are excluded.

If we bar impredicative definitions, we are unable to define many objects, such as the least upper bound of a bounded set of reals. We must thus either find an alternative way to introduce these objects, or do without them. Russell and Whitehead chose the former course, with their Axiom of Reducibility. The monograph *Das Kontinuum* was Weyl's attempt to follow the latter course: to show how much of classical mathematics could be preserved while excluding impredicative definitions.

2.2. Subsystems of Second Order Arithmetic

We are concerned in this paper with two subsystems of second order arithmetic, ACA_0 and ACA. The letters ACA stand for 'arithmetical comprehension axiom'. The system ACA_0 is investigated in great detail in Simpson [1]. These two systems are theories in the language of second order arithmetic, a language for describing natural numbers and sets of natural numbers. We now introduce this language formally.

Definition 2.1 (Language of Second Order Arithmetic). The *language of second order arithmetic* L is defined as follows.

There are two countably infinite, disjoint sets of *variables*: the *number variables* x, y, z, ..., intended to range over natural numbers; and the *set variables* X, Y, Z, ..., intended to range over sets of natural numbers.

The *terms* and *propositions* of second order arithmetic are given by the following grammar:

$$\begin{array}{lll} \text{Term} & t & ::= \quad x \mid 0 \mid \mathrm{S}t \mid t+t \mid t \cdot t \\ \text{Proposition} & \phi & ::= \quad t=t \mid t \in X \mid \bot \mid \phi \supset \phi \mid \forall x \phi \mid \forall X \phi \end{array}$$

We define $\neg$, $\wedge$, $\vee$, $\leftrightarrow$ and $\exists$ in terms of $\bot$, $\supset$ and $\forall$ as usual.

A proposition is *arithmetic* iff no set quantifier $\forall X$ occurs within it.

2.2.1. ACA_0

The system ACA_0 has been very well studied. In particular, it has played a major role in the project of Reverse Mathematics [1]. It has often been argued that ACA_0 is closely related to Weyl's foundation; for example, Feferman [3] calls it 'a modern formulation of Weyl's system', and Brown and Simpson [9] write 'ACA_0 isolates the same portion of mathematical practice which was identified as 'predicative analysis' by Herman Weyl in his famous monograph Das Kontinuum'.

It is known that ACA_0 is conservative over Peano Arithmetic (PA); a model-theoretic proof is given in Simpson [1], and a proof-theoretic proof can be given

[1] The notation here is ours, not Weyl's.

along the lines of Shoenfield [10]. A novel proof of this result shall be given in Section 6.2.

The axioms of ACA_0 are as follows:

- The *Peano axioms* — the axioms of Peano Arithmetic, minus the induction axioms:

$$\begin{gathered} \mathrm{S}x \neq 0 \\ \mathrm{S}x = \mathrm{S}y \supset x = y \\ x + 0 = x \\ x + \mathrm{S}y = \mathrm{S}(x + y) \\ x \cdot 0 = 0 \\ x \cdot \mathrm{S}y = x \cdot y + x \end{gathered}$$

- The *arithmetical comprehension axiom schema*: for every arithmetic proposition ϕ in which X does not occur free, $\exists X \forall x (x \in X \leftrightarrow \phi)$.
- The *set induction axiom*: $0 \in X \supset \forall x (x \in X \supset \mathrm{S}x \in X) \supset \forall x . x \in X$.

2.2.2. ACA

The system ACA is formed by extending ACA_0 with the full *induction axiom schema*: for every proposition ϕ,

$$[0/x]\phi \supset \forall x (\phi \supset [\mathrm{S}x/x]\phi) \supset \forall x \phi \quad .$$

An argument could be made for ACA being a better representation of the foundation in *Das Kontinuum* than ACA_0, because — as we shall argue in Section 2.3 — Weyl makes use of an induction principle that is stronger than that of ACA_0.

The system ACA has not been studied in the literature as much as ACA_0. A few facts about ACA are known: its proof-theoretic ordinal is $\varepsilon_{\varepsilon_0}$, and it can prove the consistency of ACA_0. See [11] for the proof of these results and an analysis of the set of models of ACA.

2.3. Das Kontinuum *and Subsystems of Second Order Arithmetic Compared*

There has been quite some argument over how well Weyl's foundation is captured by a subsystem of second order arithmetic. Feferman [12] has argued strongly in favour of ACA_0, or a system very like it, being a modern formulation of Weyl's system.

This argument cannot be settled formally, as Weyl did not give a formal definition of his system. However, in the authors' view, Weyl's system exceeds both ACA_0 and ACA, for the following reasons:

1. Weyl intended his system to be more than second order. He allowed the category $\mathrm{Set}\,(B)$ to be formed for *any* category B, basic or ideal. Thus, for example, we can form the categories $\mathrm{Set}\,(\mathrm{Set}\,(\mathbb{N}))$, $\mathrm{Set}\,(\mathrm{Set}\,(\mathrm{Set}\,(\mathbb{N})))$, and so forth.

2. Weyl intended the principle of induction to apply to all propositions, arithmetic or not.

We justify this by showing a place where Weyl explicitly defines a function of category $\mathrm{Set}\,(\mathrm{Set}\,(A)) \to \mathrm{Set}\,(\mathrm{Set}\,(A))$, and three places where he proves a non-arithmetic proposition by induction.

The former occurs [13, p.39] with the definition of the cardinality of a set. Weyl defines a function $d : \mathrm{Set}\,(\mathrm{Set}\,(A)) \to \mathrm{Set}\,(\mathrm{Set}\,(A))$ by

$$d(\mathscr{T}) = \{X \mid \exists x \in X. X \setminus \{x\} \in \mathscr{T}\} \enspace .$$

This function is then iterated, to form the function
$d^n(\mathscr{T}) = \{X \mid n \text{ elements may be removed from } X \text{ to form an element of } \mathscr{T}\}$. Weyl goes on to argue that $d^n(\mathscr{U})$ denotes the set of all sets with at least n elements (where $\mathscr{U}$ is the set of all subsets of A). He defines the proposition $a(n, X)$, 'X has at least n elements', by

$$a(n, X) \equiv X \in d^n(\mathscr{U})$$

Various results about this definition are later proved [13, p.55], such as:

> If X has at least $n + 1$ elements, then X has at least n elements.

This is not an arithmetic proposition (it involves quantification over X), but it is proven by induction on n.

Similarly, the non-arithmetic proposition 'If X is a subset of E and X consists of at least n elements, then E also consists of at least n elements' [13, p.56] is proven by induction, as is the lemma concerning *substitution of elements* [13, p.56]: 'If a new object [...] is substituted for one of the elements of a set X which consists of at least n elements [...], then the modified set X^* also consists of at least n elements.'

Thus, Weyl's method of defining $a(n, X)$ involves third-order sets; the application of the Principle of Iteration to third-order sets; and proof by induction of a proposition that quantifies over sets. These are all expressed by primitive constructs in $\mathrm{LTT_W}$, but not in $\mathrm{LTT_0}$ or $\mathrm{LTT_0^*}$ (we discuss this point further in Section 3.2).

When we have proven the conservativity of $\mathrm{LTT_0}$ and $\mathrm{LTT_0^*}$ over $\mathrm{ACA_0}$ and ACA respectively, we will have justified our claim that Weyl's system is stronger than $\mathrm{ACA_0}$; and, if our conjecture that $\mathrm{LTT_W}$ is stronger than $\mathrm{LTT_0^*}$ is correct, that Weyl's system is stronger than ACA.

3. Logic-Enriched Type Theories

In this section, we introduce the logic-enriched type theory $\mathrm{LTT_W}$ and the two subsystems with which we are concerned.

Logic-enriched type theories (LTTs) were introduced by Aczel and Gambino [4, 5] to study the relationship between type theories and set theories. An LTT is a formal system consisting of two parts: the *type-theory component*, which deals with *terms* and *types*; and the *logical component*, which deals with *propositions*.

3.1. $\mathrm{LTT_W}$

The system $\mathrm{LTT_W}$ is a logic-enriched type theory designed to represent the mathematical foundation given in *Das Kontinuum*. It was introduced in Adams and Luo [7, 6].

3.1.1. Type-Theoretic Component

Its type-theoretic component has the following types.

- There is a type $\mathbb{N}$ of natural numbers. 0 is a natural number; and, for any natural number N, the *successor* of N, $\mathrm{s}\,N$, is a natural number.
- For any types A and B, we may form the type $A \times B$. Its terms are pairs $(M, N)_{A\times B}$ consisting of a term M of A and a term N of B. For any term $M : A \times B$, we can construct the term $\pi_1^{A\times B}(M)$ denoting its first component, and the term $\pi_2^{A\times B}(M)$ denoting its second component.
- For any types A and B, we may form the type $A \to B$ of functions from A to B. Its terms have the form $\lambda x : A.M : B$, denoting the function which, given $N : A$, returns the term $[N/x]M : B$. Given $M : A \to B$ and $N : A$, we may construct the term $M(N)_{A\to B}$ to denote the value of the function M when applied to N.
- For any type A, we may form the type $\mathrm{Set}\,(A)$ of *sets* of terms of A. Its terms have the form $\{x : A \mid P\}$, where P is a name of a small proposition, denoting the set of all $M : A$ for which the proposition named by $[M/x]P$ is true.

We divide the types into *small* and *large* types, reflecting Weyl's division of categories into *basic* and *ideal* categories. When we introduce a set $\{x : A \mid P\}$, the proposition P may quantify over the small types, but not over the large types. The small types are defined inductively by:

- $\mathbb{N}$ is a small type.
- If A and B are small types, then $A \times B$ is a small type.

We effect this division by introducing a *type universe* U, whose terms are *names* of the small types. There is a term $\hat{\mathbb{N}} : U$ which is the name of $\mathbb{N}$; and, if $M : U$ names A and $N : U$ names B, then there is a term $M \hat{\times} N : U$ that names $A \times B$. We write $T(M)$ for the type named by M.

We can also *eliminate* $\mathbb{N}$ over any family of types; that is, if $A[x]$ is a type depending on $x : \mathbb{N}$, we can define by recursion a function f such that $f(x) : A[x]$ for all $x : \mathbb{N}$. The term

$$\mathrm{E}_{\mathbb{N}}([x]A, L, [x, y]M, N)$$

is intended to denote the value $f(N)$, where f is the function defined by recursion thus:

$$\begin{array}{lcll} f(n) & : & [n/x]A & \text{for all } n : \mathbb{N} \\ f(0) & = & L & \\ f(n+1) & = & [n/x, f(n)/y]M & \end{array}$$

Remark. We choose to label the terms

$$(M, N)_{A\times B}, \pi_1^{A\times B}(M), \pi_2^{A\times B}(M), \lambda x : A.M : B \text{ and } M(N)_{A\times B}$$

with the types A and B. This is for technical reasons only; it makes the interpretations we introduce in Section 5 easier to define. We shall often omit these labels when writing terms. We shall also often write MN for $M(N)$.

3.1.2. Logical Component

The logical component of $\mathrm{LTT_W}$ contains propositions built up as follows:

- If M and N are objects of the small type $T(L)$, then $M =_L N$ is a proposition.
- $\bot$ is a proposition.
- If ϕ and ψ are propositions, then $\phi \supset \psi$ is a proposition.
- If A is a type and ϕ a proposition, then $\forall x : A.\phi$ is a proposition.

We define the other logical connectives as follows:

$$\begin{array}{lcl}
\neg\phi & \equiv & \phi \supset \bot \\
\phi \wedge \psi & \equiv & \neg(\phi \supset \neg\psi) \\
\phi \vee \psi & \equiv & \neg\phi \supset \psi \\
\phi \leftrightarrow \psi & \equiv & (\phi \supset \psi) \wedge (\psi \supset \phi) \\
\exists x : A.\phi & \equiv & \neg\forall x : A.\neg\phi
\end{array}$$

We call a proposition ϕ *small* iff, for every quantifier $\forall x : A$ that occurs in ϕ, the type A is a small type. We wish it to be the case that, when we introduce a set of type Set (A), the proposition we use to do so must be a *small* proposition.

We achieve this by introducing a *propositional universe* 'prop', which will be the collection of names of the small propositions. We shall introduce a new judgement form $\Gamma \vdash P$ prop, denoting that P is the name of a small proposition, and rules that guarantee:

- If M and N are objects of the small type $T(L)$, then $M \hat{=}_L N$ is the name of $M =_L N$.
- $\hat{\bot}$ is the name of $\bot$.
- If P names ϕ and Q names ψ, then $P \hat{\supset} Q$ is the name of $\phi \supset \psi$.
- If $M : U$ names the small type A and P names ϕ, then $\hat{\forall} x : M.P$ names $\forall x : A.\phi$.

We denote by $V(P)$ the small proposition named by P. We shall, in the sequel, often write just 'small proposition' when we should strictly write 'name of small proposition'.

We use 'expression' to mean a type, term, small proposition or proposition. We identify expressions up to α-conversion. We denote by $[M/x]X$ the result of substituting the term M for the variable x in the expression X, avoiding variable capture.

3.1.3. Judgements and Rules of Deduction

A *context* in $\mathrm{LTT_W}$ has the form $x_1 : A_1, \ldots, x_n : A_n$, where the x_is are distinct variables and each A_i is a type. There are ten judgement forms in $\mathrm{LTT_W}$:

- $\Gamma \vdash \text{valid}$, denoting that Γ is a valid context.
- $\Gamma \vdash A \text{ type}$, denoting that A is a well-formed type under the context Γ.
- $\Gamma \vdash A = B$, denoting that A and B are equal types.
- $\Gamma \vdash M : A$, denoting that M is a term of type A.
- $\Gamma \vdash M = N : A$, denoting that M and N are equal terms of type A.
- $\Gamma \vdash P \text{ prop}$, denoting that P is a well-formed name of a small proposition.
- $\Gamma \vdash P = Q$, denoting that P and Q are equal names of small propositions.
- $\Gamma \vdash \phi \text{ Prop}$, denoting that ϕ is a well-formed proposition.
- $\Gamma \vdash \phi = \psi$, denoting that ϕ and ψ are equal propositions.
- $\Gamma \vdash \phi_1, \ldots, \phi_n \Rightarrow \psi$, denoting that the propositions $\phi_1, \ldots, \phi_n$ entail the proposition ψ.

The rules of deduction of $\mathrm{LTT_W}$ are given in full in Appendix Appendix A.1. They consist of the introduction, elimination and computation rules for the types of $\mathrm{LTT_W}$, the rules for classical predicate logic, and the following rule for performing *induction* over $\mathbb{N}$:

$$(\mathrm{Ind}_{\mathbb{N}})\dfrac{\begin{array}{cc}\Gamma, x : \mathbb{N} \vdash \phi \text{ Prop} & \Gamma \vdash N : \mathbb{N} \\ \Gamma \vdash \Phi \Rightarrow [0/x]\phi & \Gamma, x : \mathbb{N} \vdash \Phi, \phi \Rightarrow [\mathrm{s}\, x/x]\phi\end{array}}{\Gamma \vdash \Phi \Rightarrow [N/x]\phi}$$

3.2. $\mathrm{LTT_0}$ *and* $\mathrm{LTT_0^*}$

We now construct two subsystems of $\mathrm{LTT_W}$, which we shall call $\mathrm{LTT_0}$ and $\mathrm{LTT_0^*}$, that correspond to $\mathrm{ACA_0}$ and ACA respectively. These subsystems are formed by changing:

- the class of types over which $\mathbb{N}$ may be eliminated (that is, the class of types A that may occur in $\mathrm{E}_{\mathbb{N}}([x]A, L, [x, y]M, N)$;
- the class of propositions that may be proved by induction (that is, the class of propositions ϕ that may occur in an instance of $(\mathrm{Ind}_{\mathbb{N}})$).

In $\mathrm{LTT_W}$, we may eliminate $\mathbb{N}$ over any type, and any proposition may be proved by induction. We form our three subsystems by weakening these two classes, as shown in Table 1

This is achieved as follows.

	Types over which $\mathbb{N}$ may be eliminated	Propositions provable by induction
$\mathrm{LTT_W}$	all	all
LTT_0	small types	small propositions
LTT_0^*	small types	propositions involving quantification over small types and Set (N)

Table 1: Subsystems of $\mathrm{LTT_W}$

1. We construct LTT_0 by modifying $\mathrm{LTT_W}$ as follows.
   - Whenever a term $\mathrm{E}_{\mathbb{N}}([x]A, L, [x, y]M, N)$ is formed, then A must have the form $T(K)$.
   - Whenever an instance of the rule $(\mathrm{Ind}_{\mathbb{N}})$ is used, the proposition ϕ must have the form $V(P)$.
   - Whenever an instance of the rule (subst), $(\mathrm{eta}_\times)$ or $(\mathrm{eta}_\to)$ is used, the proposition ϕ must not contain a quantifier $\forall x : A$ over any type A that contains the symbol U.
   - We also add as an axiom that $\mathrm{S}M \neq 0$ for $M : \mathbb{N}$.
2. Let us say that a proposition ϕ is *analytic* iff, for every quantifier $\forall x : A$ in ϕ, A either has the form $T(M)$ or $A \equiv \mathrm{Set}\,(\mathbb{N})$. We construct LTT_0^* from LTT_0 by allowing $(\mathrm{Ind}_{\mathbb{N}})$ to be used whenever ϕ is an analytic proposition.

The formal definitions of both these systems are given in Appendices Appendix A.2 and Appendix A.3.

Remarks.

1. Peano's fourth axiom, that $\mathrm{S}M \neq_{\hat{\mathbb{N}}} 0$ for any $M : \mathbb{N}$, is provable in $\mathrm{LTT_W}$; see [6] for a proof. It is not provable in LTT_0 or LTT_0^*. This can be shown by a similar method to Smith [14] by constructing a model of LTT_0^* in which every small type is interpreted by a set that has exactly one element.
2. We can now justify further our claim in Section 2.3 that Weyl's definition of $a(n, X)$ uses the primitive concepts of $\mathrm{LTT_W}$ that are not present in either LTT_0 or LTT_0^*.
   The definitions of d and a are straightforward to formalise in $\mathrm{LTT_W}$. Given $M : U$, we have

   $$\begin{aligned} d_M &\equiv \lambda \mathscr{T} : \mathrm{Set}\,(\mathrm{Set}\,(T(M)))\,. \\ &\qquad \{X : \mathrm{Set}\,(T(M)) \mid \hat{\exists} x : M.(x \hat{\in} X \wedge X \setminus \{x\} \hat{\in} \mathscr{T})\} \\ a_M &\equiv \lambda n : \mathbb{N}.\lambda X : \mathrm{Set}\,(T(M))\,. \\ &\qquad X \in \mathrm{E}_{\mathbb{N}}([x]\mathrm{Set}\,(\mathrm{Set}\,(T(M)))\,, \mathscr{U}, [x, Y]d_M(Y), n) \end{aligned}$$

   This is not a term in either of the subsystems of $\mathrm{LTT_W}$, as it involves applying $\mathrm{E}_{\mathbb{N}}$ to the type $\mathrm{Set}\,(\mathrm{Set}\,(T(M)))$.

3. The universe U contains only the types that can be built up from $\mathbb{N}$ and $\times$. Its inclusion in LTT_0 or LTT_0^* therefore does not increase the proof-theoretic strength of the system (this will be proven in Section 5.3). This is a rare situation; in general, the inclusion of a universe raises the strength of a type theory considerably (see for example [15]). We conjecture that, if we closed U under $\rightarrow$ or $\mathrm{Set}\,(\,)$ in LTT_0 or LTT_0^*, the resulting system would not be conservative over ACA_0 or ACA respectively.
4. In Aczel and Gambino's original formulation of LTTs [4, 5], the logical component of an LTT could depend on the type theoretic component, but not *vice versa*. We have broken that restriction with the inclusion of *typed sets*: a canonical object of $\mathrm{Set}\,(A)$ has the form $\{x : A \mid P\}$ and thus depends on a small proposition P.

3.3. Embedding Second Order Systems in Logic-Enriched Type Theories

There is a translation that can naturally be defined from the language of second order arithmetic Ł into LTT_0. We map the terms of Ł to terms of type $\mathbb{N}$, first order quantifiers to quantifiers over $\mathbb{N}$, and second order quantifiers to quantifiers over $\mathrm{Set}\,(\mathbb{N})$.

Definition 3.1. We define

- for every term t of Ł, a term $\langle\!| t |\!\rangle$ of $\mathrm{LTT}_{\mathrm{W}}$;
- for every arithmetic formula ϕ of Ł, a small proposition $|\phi|$ of $\mathrm{LTT}_{\mathrm{W}}$;
- for every formula ϕ of Ł, a proposition $\langle\!| \phi |\!\rangle$ of $\mathrm{LTT}_{\mathrm{W}}$.

$$\begin{array}{rcl}
\langle\!| x_i |\!\rangle & \equiv & x_i \\
\langle\!| 0 |\!\rangle & \equiv & 0 \\
\langle\!| t' |\!\rangle & \equiv & \mathrm{s}\,\langle\!| t |\!\rangle \\
\langle\!| s+t |\!\rangle & \equiv & \langle\!| s |\!\rangle \;\texttt{plus}\; \langle\!| t |\!\rangle \\
\langle\!| s\cdot t |\!\rangle & \equiv & \langle\!| s |\!\rangle \;\texttt{times}\; \langle\!| t |\!\rangle
\end{array}$$

$$\begin{array}{rcl@{\qquad}rcl}
 & & & \langle\!| s=t |\!\rangle & \equiv & \langle\!| s |\!\rangle =_{\hat{\mathbb{N}}} \langle\!| t |\!\rangle \\
|s=t| & \equiv & \langle\!| s |\!\rangle \,\hat{=}_{\hat{\mathbb{N}}}\, \langle\!| t |\!\rangle & \langle\!| t \in X_i |\!\rangle & \equiv & \langle\!| t |\!\rangle \in_{\hat{\mathbb{N}}} X_i \\
|t \in X_i| & \equiv & \langle\!| t |\!\rangle \,\hat{\in}_{\hat{\mathbb{N}}} X_i & \langle\!| \neg\phi |\!\rangle & \equiv & \neg\,\langle\!| \phi |\!\rangle \\
|\neg\phi| & \equiv & \hat{\neg}\,|\phi| & \langle\!| \phi\supset\psi |\!\rangle & \equiv & \langle\!| \phi |\!\rangle \supset \langle\!| \psi |\!\rangle \\
|\phi\supset\psi| & \equiv & |\phi|\,\hat{\supset}\,|\psi| & \langle\!| \forall x\phi |\!\rangle & \equiv & \forall x : \mathbb{N}.\, \langle\!| \phi |\!\rangle \\
|\forall x\phi| & \equiv & \hat{\forall} x : \hat{\mathbb{N}}.\, |\phi| & \langle\!| \forall X\phi |\!\rangle & \equiv & \forall X : \mathrm{Set}\,(\mathbb{N})\,.\, \langle\!| \phi |\!\rangle
\end{array}$$

where

$$\begin{array}{rcl}
M \;\texttt{plus}\; N & \equiv & \mathrm{E}_{\mathbb{N}}([x]T(\hat{\mathbb{N}}), M, [x,y]\,\mathrm{s}\,y, N) \\
M \;\texttt{times}\; N & \equiv & \mathrm{E}_{\mathbb{N}}([x]T(\hat{\mathbb{N}}), 0, [x,y]y \;\texttt{plus}\; M, N)
\end{array}$$

It is straightforward to show that this translation is sound, in the following sense:

Theorem 3.2 *Let* Γ *be the context* $x_1 : \mathbb{N}, \ldots, x_m : \mathbb{N}, X_1 : \mathrm{Set}\,(\mathbb{N})\,, \ldots, X_n : \mathrm{Set}\,(\mathbb{N})$*. Let* $\mathrm{FV}\,(t) \subseteq \{x_1, \ldots, x_m\}$*, and* $\mathrm{FV}\,(\phi) \subseteq \{x_1, \ldots, x_m, X_1, \ldots, X_n\}$*.*

1. $\Gamma \vdash \llparenthesis t \rrparenthesis : \mathbb{N}$ *and* $\Gamma \vdash \llparenthesis \phi \rrparenthesis$ Prop*.*
2. *If* ϕ *is arithmetic, then* $\Gamma \vdash |\phi|$ prop *and* $\Gamma \vdash V(|\phi|) = \llparenthesis \phi \rrparenthesis$*.*
3. *If* $\mathrm{ACA}_0 \vdash \phi$*, then* $\Gamma \vdash\Rightarrow \llparenthesis \phi \rrparenthesis$ *in* LTT_0*.*
4. *If* $\mathrm{ACA} \vdash \phi$*, then* $\Gamma \vdash\Rightarrow \llparenthesis \phi \rrparenthesis$ *in* LTT_0^**.*

Proof. Parts 1 and 2 are proven straightforwardly by induction on t and ϕ.

For part 3, it is sufficient to prove the case where ϕ is an axiom of ACA_0. The case of the Peano axioms is straightforward.

For the arithmetical comprehension axiom schema, let ϕ be an arithmetic formula in which X does not occur free. We have

$$\begin{array}{rcll} \Gamma & \vdash & \Rightarrow \forall x : \mathbb{N}(V(|\phi|) \leftrightarrow \llparenthesis \phi \rrparenthesis) & \text{(using part 1)} \\ \therefore \Gamma & \vdash & \Rightarrow \forall x : \mathbb{N}(x \in \{x : \mathbb{N} \mid |\phi|\} \leftrightarrow \llparenthesis \phi \rrparenthesis) & \\ \therefore \Gamma & \vdash & \Rightarrow \exists X : \mathrm{Set}\,(\mathbb{N})\,.\forall x : \mathbb{N}(x \in X \leftrightarrow \llparenthesis \phi \rrparenthesis) & \end{array}$$

as required.

The set induction axiom is shown to be provable using $(\mathrm{Ind}_{\mathbb{N}})$.

For part 4, it is sufficient to show that every instance of the full induction axiom schema is provable in LTT_0^*. This is easy to do using $(\mathrm{Ind}_{\mathbb{N}})$, as $\llparenthesis \phi \rrparenthesis$ is always an analytic proposition.

Corollary 3.2.1 $\mathrm{LTT}_{\mathrm{W}}$ *is strictly stronger than* ACA_0*. In fact,* $\mathrm{LTT}_{\mathrm{W}}$ *can prove the consistency of* ACA_0*.*

Proof. As ACA_0 is conservative over Peano Arithmetic [1], its proof-theoretic ordinal is ϵ_0. The proof-theoretic ordinal of ACA is ϵ_{ϵ_0} [11, 16]. Therefore, ACA can prove the consistency of ACA_0; hence, so can LTT_0^*; hence, so can $\mathrm{LTT}_{\mathrm{W}}$.

Our aim in this paper is to prove the converse to Theorem 3.2 parts 3 and 4: that, whenever $\Gamma \vdash\Rightarrow \llparenthesis \phi \rrparenthesis$ in LTT_0 or LTT_0^*, then ϕ is provable in the corresponding subsystem of second order arithmetic.

4. Conservativity of $\mathbf{T_2}$ over $\mathbf{ACA_0}$

We shall now define the system T_2, which is a subsystem of LTT_0. We can think of T_2 as the second order fragment of LTT_0; that is, the part of LTT_0 that has just the two types $\mathbb{N}$ and $\mathrm{Set}\,(\mathbb{N})$.

The translation $\llparenthesis\ \rrparenthesis$ given in the previous section is in fact a sound translation of ACA_0 into T_2. In this section, we shall prove that this translation is conservative; that is, if $\llparenthesis \phi \rrparenthesis$ is provable in T_2, then ϕ is a theorem of ACA_0.

The syntax of T_2 is given by the following grammar

$$\begin{array}{llll} \text{Type} & A & ::= & \mathbb{N} \mid \mathrm{Set}\,(\mathbb{N}) \\ \text{Term} & M & ::= & x \mid 0 \mid \mathrm{s}\, M \mid \mathrm{R}(M, [x,x]M, M) \mid \{x : \mathbb{N} \mid P\} \\ \text{Small Proposition} & P & ::= & M \hat{=}_{\mathbb{N}} M \mid \hat{\bot} \mid P \hat{\supset} P \mid \hat{\forall} x : \hat{\mathbb{N}}.P \mid M \hat{\in}_{\mathbb{N}} M \\ \text{Proposition} & \phi & ::= & M =_{\mathbb{N}} M \mid \bot \mid \phi \supset \phi \mid \forall x : A.\phi \mid V(P) \end{array}$$

$$\frac{\Gamma, x : \mathbb{N} \vdash P \text{ Prop}}{\Gamma \vdash \hat{\forall} x : \hat{\mathbb{N}}.P \text{ Prop}} \qquad \frac{\Gamma, x : \mathbb{N} \vdash P = Q}{\Gamma \vdash (\hat{\forall} x : \hat{\mathbb{N}}.P) = (\hat{\forall} x.\hat{\mathbb{N}}.Q)}$$

$$\frac{\Gamma, x : \mathbb{N} \vdash P \text{ Prop}}{\Gamma \vdash V(\hat{\forall} x : \hat{\mathbb{N}}.P) = \forall x : \mathbb{N}.V(P)}$$

Figure 1: Rules of Deduction for Small Universal Quantification in T_2

$$\frac{\Gamma \vdash \text{valid}}{\Gamma \vdash \mathbb{N} \text{ type}} \qquad \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash 0 : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathrm{s}\, M : \mathbb{N}} \qquad \frac{\Gamma \vdash M = M' : \mathbb{N}}{\Gamma \vdash \mathrm{s}\, M = \mathrm{s}\, M' : \mathbb{N}}$$

$$\frac{\begin{array}{c}\Gamma \vdash L : \mathbb{N} \\ \Gamma, x : \mathbb{N}, y : \mathbb{N} \vdash M : \mathbb{N} \\ \Gamma \vdash N : \mathbb{N}\end{array}}{\Gamma \vdash \mathrm{R}(L, [x,y]M, N) : \mathbb{N}} \qquad \frac{\begin{array}{c}\Gamma \vdash L = L' : \mathbb{N} \\ \Gamma, x : \mathbb{N}, y : \mathbb{N} \vdash M = M' : \mathbb{N} \\ \Gamma \vdash N = N' : \mathbb{N}\end{array}}{\Gamma \vdash \mathrm{R}(L, [x,y]M, N) = \mathrm{R}(L', [x,y]M', N') : \mathbb{N}}$$

$$\frac{\begin{array}{c}\Gamma \vdash L : \mathbb{N} \\ \Gamma, x : \mathbb{N}, y : \mathbb{N} \vdash M : \mathbb{N}\end{array}}{\Gamma \vdash \mathrm{R}(L, [x,y]M, 0) = L : \mathbb{N}} \qquad \frac{\begin{array}{c}\Gamma \vdash L : \mathbb{N} \\ \Gamma, x : \mathbb{N}, y : \mathbb{N} \vdash M : \mathbb{N} \\ \Gamma \vdash N : \mathbb{N}\end{array}}{\begin{array}{l}\Gamma \vdash \mathrm{R}(L, [x,y]M, \mathrm{s}\, N) \\ \quad = [N/x, \mathrm{R}(L, [x,y]M, N)/y]M : \mathbb{N}\end{array}}$$

$$(\mathrm{Ind}_{\mathbb{N}})\ \frac{\begin{array}{cc}\Gamma, x : \mathbb{N} \vdash P \text{ Prop} & \Gamma \vdash N : \mathbb{N} \\ \Gamma \vdash \Phi \Rightarrow V([0/x]P) & \Gamma \vdash \Phi, V(P) \Rightarrow V([\mathrm{s}\, x/x]P)\end{array}}{\Gamma \vdash \Phi \Rightarrow V([N/x]P)}$$

Figure 2: Rules of Deduction for Natural Numbers in T_2

The rules of deduction of T_2 are:

1. the structural rules for LTTs as given in Appendix Appendix A.1.1;
2. the rules for predicate logic as given in Appendix Appendix A.1.7;
3. the rules for the propositional universe as given in Appendix Appendix A.1.8, with the rules for universal quantification replaced with the rules in Figure 1;
4. the rules for equality given in Appendix Appendix A.1.9, restricted to the type $\mathbb{N}$;
5. the rules for sets given in Appendix Appendix A.1.5, restricted to the type $\mathbb{N}$;
6. the rules for natural numbers given in Figure 2.

Note. T_2 does not contain the universe U. The symbol $\hat{\mathbb{N}}$ therefore is not a term in T_2, and cannot occur on its own, but only as part of a small proposition

$\hat{\forall} x : \hat{\mathbb{N}}.P$.

In LTT_0, we could define functions by recursion into any small type; in T_2, we can only define by recursion functions from $\mathbb{N}$ to $\mathbb{N}$. This is achieved by the constructor R. The term $\mathrm{R}(L, [x, y]M, N)$ is intended to denote the value $f(N)$, where $f : \mathbb{N} \to \mathbb{N}$ is defined by recursion thus:

$$\begin{array}{lcl} f(0) & = & L \\ f(n+1) & = & [n/x, f(n)/y]M \end{array}$$

The system T_2 may be considered a subsystem of LTT_0 if we identify $\mathrm{R}(L, [x, y]M, N)$ with $\mathrm{E}_{\mathbb{N}}([x]\hat{\mathbb{N}}, L, [x, y]M, N)$; $M =_{\mathbb{N}} N$ with $M =_{\hat{\mathbb{N}}} N$; and $M \hat{=}_{\mathbb{N}} N$ with $M \hat{=}_{\hat{\mathbb{N}}} N$.

The translation given in Section 3.3 is a sound translation from ACA_0 into T_2.

Theorem 4.1 *Let* Γ *and* ϕ *be as in Theorem 3.2. If* $\mathrm{ACA}_0 \vdash \phi$*, then* $\Gamma \vdash\Rightarrow \langle\!|\phi|\!\rangle$ *in* T_2*.*

PROOF. Similar to the proof of Theorem 3.2(3).

We now wish to show that the converse holds.

We shall do this by defining the following translation Φ from T_2 to ACA_0. Let

$$\Gamma \equiv x_1 : \mathbb{N}, \ldots, x_n : \mathbb{N}, X_1 : \mathrm{Set}\,(\mathbb{N}), \ldots, X_m : \mathrm{Set}\,(\mathbb{N}) \enspace .$$

We shall define:

1. whenever $\Gamma \vdash M : \mathbb{N}$, an arithmetic formula $t \ulcorner = M \urcorner$ such that

$$\mathrm{ACA}_0 \vdash \exists! x . x \ulcorner = M \urcorner \enspace .$$

   The intention is that M is interpreted as the unique number x for which $x \ulcorner = M \urcorner$ is true.
2. whenever $\Gamma \vdash M : \mathrm{Set}\,(\mathbb{N})$, an arithmetic formula $t \ulcorner \in M \urcorner$ such that

$$\mathrm{ACA}_0 \vdash \exists X \forall x (x \in X \leftrightarrow x \ulcorner \in M \urcorner) \enspace ,$$

   The intention is that M is interpreted as the unique set X whose members are the numbers x such that $x \ulcorner \in M \urcorner$ is true.
3. for every small proposition P such that $\Gamma \vdash P$ prop, an arithmetic formula $\ulcorner P \urcorner$.
4. for every proposition ϕ such that $\Gamma \vdash \phi$ Prop, a formula $\ulcorner \phi \urcorner$.

The definition is given in Figure 3.

Numbers.

$$
\begin{aligned}
t \ulcorner = x_i \urcorner &\equiv t = x_i \\
t \ulcorner = 0 \urcorner &\equiv t = 0 \\
t \ulcorner = \mathrm{s}\, M \urcorner &\equiv \exists x (x \ulcorner = M \urcorner \wedge t = \mathrm{S} x) \\
t \ulcorner = \mathrm{R}(L, [u, v] M, N) \urcorner &\equiv \exists n . \exists s \in \mathbf{Seq} (n \ulcorner = N \urcorner \wedge (n, t) \in s \\
&\qquad \wedge \forall l ((0, l) \in s \supset l \ulcorner = L \urcorner) \\
&\qquad \wedge \forall u \forall z ((\mathrm{S} u, z) \in s \supset \exists v ((u, v) \in s \wedge z \ulcorner = M \urcorner)))
\end{aligned}
$$

Sets.

$$
\begin{aligned}
t \ulcorner \in X_i \urcorner &\equiv t \in X_i \\
t \ulcorner \in \{ x : \mathbb{N} \mid P \} \urcorner &\equiv [t/x] \ulcorner P \urcorner
\end{aligned}
$$

Small Propositions.

$$
\begin{aligned}
\ulcorner M \hat{=}_{\mathbb{N}} N \urcorner &\equiv \exists x (x \ulcorner = M \urcorner \wedge x \ulcorner = N \urcorner) \\
\ulcorner \hat{\bot} \urcorner &\equiv \bot \\
\ulcorner P \hat{\supset} Q \urcorner &\equiv \ulcorner P \urcorner \supset \ulcorner Q \urcorner \\
\ulcorner \hat{\forall} x : \mathbb{N} . P \urcorner &\equiv \forall x \ulcorner P \urcorner \\
\ulcorner M \hat{\in}_{\mathbb{N}} N \urcorner &\equiv \exists x (x \ulcorner = M \urcorner \wedge x \ulcorner \in N \urcorner)
\end{aligned}
$$

Propositions.

$$
\begin{aligned}
\ulcorner M =_{\mathbb{N}} N \urcorner &\equiv \exists x (x \ulcorner = M \urcorner \wedge x \ulcorner = N \urcorner) \\
\ulcorner \bot \urcorner &\equiv \bot \\
\ulcorner \phi \supset \psi \urcorner &\equiv \ulcorner \phi \urcorner \supset \ulcorner \psi \urcorner \\
\ulcorner \forall x : \mathbb{N} . \phi \urcorner &\equiv \forall x \ulcorner \phi \urcorner \\
\ulcorner \forall X : \mathrm{Set}\, (\mathbb{N})\, . \phi \urcorner &\equiv \forall X \ulcorner \phi \urcorner \\
\ulcorner V(P) \urcorner &\equiv \ulcorner P \urcorner
\end{aligned}
$$

Figure 3: Interpretation of T_2 in ACA_0

Remark. To interpret a term of the form $\mathrm{R}(L, [u, v]M, N)$, we make use of a standard technique for defining functions by recursion in ACA_0. We are assuming we have defined in ACA_0 a pairing function (m, n) on the natural numbers, and a coding of finite sequences of numbers as numbers, with **Seq** the set of all codes of sequences, and the formula $n \in s$ expressing that n is a member of the sequence coded by s. (For more details, see [1, II.3].)

Speaking informally, the formula $t \ulcorner = R(L, [u, v]M, N)\urcorner$ expresses that (N, t) is a member of a sequence s, and that the members of this sequence s must be

$$(0, \mathrm{R}(L, [u, v]M, 0)), \quad (1, \mathrm{R}(L, [u, v]M, 1)), \quad \cdots, \quad (k, R(L, [u, v]M, k))$$

up to some k, in some order. It follows that $t = \mathrm{R}(L, [u, v]M, N)$.

The following theorem shows that the translation in Figure 3 is sound.

Theorem 4.2 (Soundness) 1. *If* $\Gamma \vdash M : \mathbb{N}$ *then* $\mathrm{ACA}_0 \vdash \exists! x . x \ulcorner = M \urcorner$.

2. *If* $\Gamma \vdash M = M' : \mathbb{N}$ *then* $\mathrm{ACA}_0 \vdash \exists x (x \ulcorner = M \urcorner \wedge x \ulcorner = M' \urcorner)$.
3. *If* $\Gamma \vdash M : \mathrm{Set}\,(\mathbb{N})$ *then* $\mathrm{ACA}_0 \vdash \exists! X \forall x (x \in X \leftrightarrow x \ulcorner \in M \urcorner)$.
4. *If* $\Gamma \vdash M = N : \mathrm{Set}\,(\mathbb{N})$ *then* $\mathrm{ACA}_0 \vdash \forall x (x \ulcorner \in M \urcorner \leftrightarrow x \ulcorner \in N \urcorner)$.
5. *If* $\Gamma \vdash P = Q$ *then* $\mathrm{ACA}_0 \vdash \ulcorner P \urcorner \leftrightarrow \ulcorner Q \urcorner$.
6. *If* $\Gamma \vdash \phi = \psi$ *then* $\mathrm{ACA}_0 \vdash \ulcorner \phi \urcorner \leftrightarrow \ulcorner \psi \urcorner$.
7. *If* $\Gamma \vdash \phi_1, \ldots, \phi_n \Rightarrow \psi$ *then* $\mathrm{ACA}_0 \vdash \ulcorner \phi_1 \urcorner \supset \cdots \supset \ulcorner \phi_n \urcorner \supset \ulcorner \psi \urcorner$.

Proof. We need the following two results first.

1. For any term M such that $x, y \notin \mathrm{FV}\,(M)$,

$$\mathrm{ACA}_0 \vdash x \ulcorner = M \urcorner \supset y \ulcorner = M \urcorner \supset x = y \ .$$

   This is proven by induction on M.

2. Given a term N such that $x \notin \mathrm{FV}\,(N)$, the following are all theorems of ACA_0:

$$x \ulcorner = N \urcorner \supset (y \ulcorner = [N/x]M \urcorner \leftrightarrow y \ulcorner = M \urcorner) \tag{2}$$

$$x \ulcorner = N \urcorner \supset (y \ulcorner \in [N/x]M \urcorner \leftrightarrow y \ulcorner \in M \urcorner) \tag{3}$$

$$\forall x (x \in X \leftrightarrow x \ulcorner \in N \urcorner) \supset (y \ulcorner \in [N/X]M \urcorner \leftrightarrow y \ulcorner \in M \urcorner) \tag{4}$$

$$x \ulcorner = N \urcorner \supset (\ulcorner [N/x]P \urcorner \leftrightarrow \ulcorner P \urcorner) \tag{5}$$

$$\forall x (x \in X \leftrightarrow x \ulcorner \in N \urcorner) \supset (\ulcorner [N/X]P \urcorner \leftrightarrow \ulcorner P \urcorner) \tag{6}$$

$$x \ulcorner = N \urcorner \supset (\ulcorner [N/x]\phi \urcorner \leftrightarrow \ulcorner \phi \urcorner) \tag{7}$$

$$\forall x (x \in X \leftrightarrow x \ulcorner \in N \urcorner) \supset (\ulcorner [N/x]\phi \urcorner \leftrightarrow \ulcorner \phi \urcorner) \tag{8}$$

   These are proven by induction on M, P or ϕ. Formulas (3)–(6) must be proven simultaneously.

The seven parts of the theorem are now proven simultaneously by induction on derivations. We deal with one case here: the rule

$$\frac{\Gamma \vdash L : \mathbb{N} \quad \Gamma, u : \mathbb{N}, v : \mathbb{N} \vdash M : \mathbb{N} \\ \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \mathrm{R}(L, [u, v]M, \mathrm{s}\, N) = [N/u, \mathrm{R}(L, [u, v]M, N)/v]M : \mathbb{N}}$$

We reason in ACA_0. By the induction hypothesis, there exist l and n such that $l \ulcorner = L \urcorner$, $n \ulcorner = N \urcorner$. Further,

$$\forall u \forall v \exists m . m \ulcorner = M \urcorner \ .$$

The following formula can be proven by induction on z:

$$\begin{aligned} \forall z \exists w \exists s \in \mathbf{Seq}((z, w) \in s & \\ & \wedge \forall l((0, l) \in s \supset l \ulcorner = L \urcorner) \\ & \wedge \forall u \forall z((\mathrm{S}u, z) \in s \supset \exists v((u, v) \in s \wedge z \ulcorner = M \urcorner))) \end{aligned}$$

Now, let n be the unique number such that $n \ulcorner = N \urcorner$. There exist m, p such that (n, m) and $(\mathrm{S}n, p)$ are members of such a sequence s. It follows that

$$p \ulcorner = \mathrm{R}(L, [u, v]M, \mathrm{s}\, n) \urcorner,\ m \ulcorner = \mathrm{R}(L, [u, v]M, n) \urcorner,\ [n/u, m/v](p \ulcorner = M \urcorner) \ .$$

Hence, by (2), we have

$$p \ulcorner = \mathrm{R}(L, [u, v]M, \mathrm{s}\, N) \urcorner, \qquad p \ulcorner = [N/u, \mathrm{R}(L, [u, v]M, N)/v]M \urcorner$$

as required.

Conservativity shall follow from the following theorem, which states that the mapping $\ulcorner\ \urcorner$ is a left-inverse to the mapping $\langle\!|\ |\!\rangle$ from ACA_0 to T_2, up to logical equivalence.

Theorem 4.3

1. *For every term t of ACA_0, we have $\mathrm{ACA}_0 \vdash t \ulcorner = \langle\!| t |\!\rangle \urcorner$.*
2. *For every arithmetic proposition ϕ of ACA_0, we have $\mathrm{ACA}_0 \vdash \phi \leftrightarrow \ulcorner |\phi| \urcorner$.*
3. *For every proposition ϕ of ACA_0, we have $\mathrm{ACA}_0 \vdash \phi \leftrightarrow \ulcorner \langle\!| \phi |\!\rangle \urcorner$.*

Proof. The proof of each of these statements is a straightforward induction. We deal with one case here: the case $t \equiv t_1 + t_2$. We reason in ACA_0. The induction hypothesis gives

$$t_1 \ulcorner = \langle\!| t_1 |\!\rangle \urcorner, \qquad t_2 \ulcorner = \langle\!| t_2 |\!\rangle \urcorner$$

and we must show $t_1 + t_2 \ulcorner = \langle\!| t_1 |\!\rangle\ \mathtt{plus}\ \langle\!| t_2 |\!\rangle \urcorner$, i.e.

$$\begin{aligned} \exists n . \exists r \in \mathbf{Seq}(n \ulcorner = \langle\!| t_2 |\!\rangle \urcorner \wedge (n, t_1 + t_2) \in r & \\ & \wedge \forall l((0, l) \in r \supset l \ulcorner = \langle\!| t_1 |\!\rangle \urcorner) \\ & \wedge \forall x \forall z((\mathrm{S}x, z) \in r \supset \exists y((x, y) \in r \wedge z = \mathrm{S}y))) \end{aligned}$$

We prove the following by induction on b:

$$\begin{aligned} \forall a, b . \exists r \in \mathbf{Seq}((b, a + b) \in r & \\ & \wedge \forall l((0, l) \in r \supset l = a) \\ & \wedge \forall x, z((\mathrm{S}x, z) \in r \supset \exists y((x, y) \in r \wedge z = \mathrm{S}y))) \end{aligned}$$

The desired proposition follows by instantiating a with t_1 and b with t_2.

Corollary 4.3.1 (Conservativity of T_2 over ACA_0) *For any formula ϕ of ACA_0, if $\Gamma \vdash \Rightarrow \llparenthesis \phi \rrparenthesis$ in T_2, then $\mathrm{ACA}_0 \vdash \phi$.*

Proof. By the Soundness Theorem, we have that $\mathrm{ACA}_0 \vdash \ulcorner \llparenthesis \phi \rrparenthesis \urcorner$. By Theorem 4.3, we have $\mathrm{ACA}_0 \vdash \phi \leftrightarrow \ulcorner \llparenthesis \phi \rrparenthesis \urcorner$. Therefore, $\mathrm{ACA}_0 \vdash \phi$.

5. Conservativity of $\mathbf{LTT_0}$ over $\mathbf{ACA_0}$

In this section, we shall prove that LTT_0 is conservative over T_2. This shall complete the proof that LTT_0 is conservative over ACA_0.

We shall do this by defining a number of subsystems of LTT_0 as shown in the diagram:

$$\mathrm{T}_2 \hookrightarrow \mathrm{T}_\omega \hookrightarrow \mathrm{T}_\omega U \hookrightarrow \mathrm{LTT}_0 \ .$$

For each of these inclusions $A \hookrightarrow B$, we shall prove that A is a conservative subsystem of B; that is, for every judgement $\mathcal{J}$ in the language of A, if $\mathcal{J}$ is derivable in B then $\mathcal{J}$ is derivable in A. This shall sometimes involve constructing yet more subsystems in between A and B, and proving that all these inclusions are conservative.

Intuitively, each subsystem deals with a subset of the types of LTT_0.

- T_2 has only two types, $\mathbb{N}$ and $\mathrm{Set}\,(\mathbb{N})$.
- The types of T_ω are all the types that can be built up from $\mathbb{N}$ using $\times$, $\rightarrow$ and $\mathrm{Set}\,()$.
- The types of $\mathrm{T}_\omega U$ are the types of T_ω, together with the universe U. (The constructors $\times$, $\rightarrow$ and $\mathrm{Set}\,()$ may *not* be applied to U in $\mathrm{T}_\omega U$.)

The formal definitions of these systems shall be given in the sections to come.

5.1. Digression — Informal Explanation of Proof Technique

Before proceeding with the technical details of the proof, we shall explain the informal ideas behind the technique we use to prove LTT_0 conservative over T_2. The system LTT_0 is formed from T_2 by adding products, function types, types of sets, and the universe U. Intuitively, none of these should increase the power of the system.

We can see this most clearly in the case of products. Speaking generally, let S be any type system, and let T be formed by adding product types to S. Then T should have no more expressive power than S, because we can envisage a translation from T to S:

- wherever a variable $z : A \times B$ occurs, replace it with two variables $x : A, y : B$;
- wherever a term of type $A \times B$ occurs, replace it with two terms, one of type A and one of type B.

As long as the only way of introducing terms of type $A \times B$ is the constructor $(\; , \;)$, we should always be able to find the two S-terms of types A and B that correspond to any T-term of type $A \times B$. (This would however not be possible if (say) we could eliminate $\mathbb{N}$ over $A \times B$ in T.)

In brief:

- the terms of type $A \times B$ can be interpreted as pairs $\langle M, N \rangle$ where $M : A$ and $N : B$.

Similarly,

- the terms of type $A \to B$ can be interpreted as pairs $\langle x, M \rangle$ where $x : A \vdash M : B$;
- the terms of type $\mathrm{Set}\,(A)$ can be interpreted as pairs $\langle x, P \rangle$ where $x : A \vdash P$ prop.

Our proof relies on making these intuitive ideas formal.

These ideas show us how we might be able to remove types $A \to B$ that involve only one use of the arrow, but they do not show us how to handle types of the form $(A \to B) \to C$. Let us take another example: let S be a typing system without function types, and let T be formed from S by adding function types. Let us define the *depth* of a type A, $d(A)$ by:

- the depth of each type in S is 0;
- $d(A \to B) = \max(d(A), d(B)) + 1$.

Then we have seen how to interpret types of depth 1 in terms of types of depth 0. More generally, we can interpret types of depth $n + 1$ in terms of types of depth n.

This shows us how to complete the proof. We introduce an infinite sequence of subsystems of T:

$$S = \mathcal{A}_0 \hookrightarrow \mathcal{A}_1 \hookrightarrow \mathcal{A}_2 \hookrightarrow \cdots T$$

where, in $\mathcal{A}_n$, only types of depth $\leq n$ may occur. We build an interpretation of $\mathcal{A}_{n+1}$ out of the terms of $\mathcal{A}_n$: every type of $\mathcal{A}_n$ is interpreted as itself; the types $A \to B$ of depth $n + 1$ are interpreted as the set of pairs $\langle x, M \rangle$ where $x : A \vdash M : B$ in $\mathcal{A}_n$.

Using these interpretations, we can prove each $\mathcal{A}_{n+1}$ conservative over $\mathcal{A}_n$, and hence T conservative over S. With these intuitive ideas to guide us, we return to the proof development.

5.2. T_ω *is Conservative over* T_2

We shall now define the system T_ω to be T_2 extended with pairs, functions and sets over all types, and prove that T_ω is conservative over T_2.

Definition 5.1 (T_ω). The LTT T_ω is defined as follows.

The grammar of T_ω is the grammar of T_2 extended with

$$\begin{array}{llcl} \text{Type} & A & ::= & \cdots \mid A \times A \mid A \to A \mid \mathrm{Set}\,(A) \\ \text{Term} & M & ::= & \cdots \mid (M, M)_{A \times A} \mid \pi_1^{A \times A}(M) \mid \pi_2^{A \times A}(M) \mid \\ & & & \lambda x : A.M : A \mid M(M)_{A \to A} \mid \{x : A \mid P\} \\ \text{Small Proposition} & P & ::= & \cdots \mid M \hat{\in}_A M \end{array}$$

The rules of deduction of T_ω are the rules of deduction of T_2, together with the rules for pairs (Appendix Appendix A.1.3), function types (Appendix Appendix A.1.4) and typed sets (Appendix Appendix A.1.5).

Note that the type-theory component T_ω is non-dependent: a term can never occur in a type. As a consequence, we have

Lemma 5.2 *If $\Gamma \vdash A = B$ in T_ω then $A \equiv B$.*

Proof. Induction on derivations.

To prove that T_ω is conservative over T_2, we shall define an infinite sequence of subsystems of T_ω, and prove that each is conservative over the previous subsystem, and that the smallest is conservative over T_2.

$$\mathrm{T}_2 \hookrightarrow \mathcal{A}_1 \hookrightarrow \mathcal{A}_2 \hookrightarrow \cdots \mathrm{T}_\omega$$

We define the *depth* of a type of T_ω as follows.

Definition 5.3. Define the *depth* $d(A) < \omega$ of a type A of T_ω by

$$\begin{array}{lcll} d(\mathbb{N}) & = & 0 & \\ d(A \times B) & = & \max(d(A), d(B)) + 1 & \\ d(A \to B) & = & \max(d(A), d(B)) + 1 & \\ d(\mathrm{Set}\,(\mathbb{N})) & = & 0 & \\ d(\mathrm{Set}\,(A)) & = & d(A) + 1 & \qquad (A \not\equiv \mathbb{N}) \end{array}$$

Note that the types of T_2 are exactly the types of depth 0.

For $n \geq 1$, we shall define $\mathcal{A}_n$ to be the fragment of T_ω that deals only with types of depth $\leq n$.

Definition 5.4 ($\mathcal{A}_n$). Let $n \geq 0$. By a *type* (term, small proposition, proposition, context, judgement) *of* $\mathcal{A}_n$, we mean a type (term, small proposition, proposition, context, judgement) of T_ω that does not contain, as a subexpression, any type of depth $> n$.

We say a judgement $\mathcal{J}$ of $\mathcal{A}_n$ is *derivable* in $\mathcal{A}_n$ iff there exists a derivation of $\mathcal{J}$ in T_ω consisting solely of judgements of $\mathcal{A}_n$; that is, a derivation of $\mathcal{J}$ in which no type of depth $> n$ occurs. We write $\Gamma \vdash_n \mathcal{J}$ iff the judgement $\Gamma \vdash \mathcal{J}$ is derivable in $\mathcal{A}_n$.

Note that the types of $\mathcal{A}_n$ are exactly the types of depth $\leq n$. Note also that $\mathcal{A}_0$ is just the system T_2.

We shall prove that $\mathcal{A}_{n+1}$ is conservative over $\mathcal{A}_n$. The proof shall involve defining an *interpretation* of $\mathcal{A}_{n+1}$ in terms of the expressions of $\mathcal{A}_n$. For the rest of this section, fix $n \geq 0$, and fix a context Δ of $\mathcal{A}_n$ such that $\Delta \vdash_n$ valid.

Definition 5.5 (Interpretation of Types). For the purposes of this definition, an 'object' is either a term of $\mathcal{A}_n$, or a pair of terms of $\mathcal{A}_n$.

For every type A of $\mathcal{A}_{n+1}$, we define the set of objects $[\![A]\!]_\Delta$, and an equivalence relation $\sim^A_\Delta$ on this set, as follows.

If $d(A) \leq n$, then

$$\begin{array}{lcl} [\![A]\!]_\Delta & = & \{M \mid \Delta \vdash_n M : A\} \\ M \sim^A_\Delta N & \Leftrightarrow & \Delta \vdash_n M = N : A \end{array}$$

Otherwise,

$$\begin{array}{lcl} [\![A \times B]\!]_\Delta & = & \{\langle M, N \rangle \mid \Delta \vdash_n M : A, \Delta \vdash_n N : B\} \\ \langle M, N \rangle \sim^{A \times B}_\Delta \langle M', N' \rangle & \Leftrightarrow & \Delta \vdash_n M = M' : A \wedge \Delta \vdash_n N = N' : B \\ & & \\ [\![A \to B]\!]_\Delta & = & \{\langle x, M \rangle \mid \Delta, x : A \vdash_n M : B\} \\ \langle x, M \rangle \sim^{A \to B}_\Delta \langle x, M' \rangle & \Leftrightarrow & \Delta, x : A \vdash_n M = M' : B \\ & & \\ [\![\mathrm{Set}\,(A)]\!]_\Delta & = & \{\langle x, P \rangle \mid \Delta, x : A \vdash_n P \text{ prop}\} \\ \langle x, P \rangle \sim^{\mathrm{Set}(A)}_\Delta \langle x, P' \rangle & \Leftrightarrow & \Delta, x : A \vdash_n P = P' \end{array}$$

We identify the elements of $[\![A \to B]\!]_\Delta$ and $[\![\mathrm{Set}\,(A)]\!]_\Delta$ up to α-conversion; that is, we identify $\langle x, M \rangle$ with $\langle y, [y/x]M \rangle$ if y is not free in M.

We define the operations Π_1, Π_2 and @ on these objects as follows.

$$\begin{array}{lcl} \Pi_1(\langle M, N \rangle) & \equiv & M \\ \Pi_2(\langle M, N \rangle) & \equiv & N \\ \langle x, M \rangle @ N & \equiv & [N/x]M \\ \langle x, P \rangle @ N & \equiv & [N/x]P \end{array}$$

$\Pi_1(X)$ and $\Pi_2(X)$ are undefined if X is not a pair. $X@Y$ is undefined if X does not have the form $\langle x, Z \rangle$, or if Y is not a term.

The intention is that we will interpret the terms of type A as members of the set $[\![A]\!]_\Delta$, with equal terms being interpreted as $\sim^A_\Delta$-equivalent members

Definition 5.6 (Valuation). Let $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$ be a context of $\mathcal{A}_{n+1}$. A Δ-*valuation* of Γ is a function v on $\{x_1, \ldots, x_n\}$ such that

$$v(x_i) \in [\![A_i]\!]_\Delta \qquad (i = 1, \ldots, n) \ .$$

Definition 5.7 (Interpretation of Terms). Given a term M of $\mathcal{A}_{n+1}$ and a function v whose domain includes $\mathrm{FV}\,(M)$, we define the object $\llparenthesis M \rrparenthesis^v_A$ as follows.

$$\begin{array}{lcl}
\llparenthesis x \rrparenthesis^v & = & v(x) \\
\llparenthesis 0 \rrparenthesis^v & \equiv & 0 \\
\llparenthesis \mathrm{s}\, M \rrparenthesis^v & \simeq & \mathrm{s} \llparenthesis M \rrparenthesis^v \\
\llparenthesis R(L, [x,y]M, N) \rrparenthesis^v & \simeq & R(\llparenthesis L \rrparenthesis^v, [x,y]\llparenthesis M \rrparenthesis^{v[x:=x, y:=y]}, \llparenthesis N \rrparenthesis^v) \\
\llparenthesis (M, N)_{A \times B} \rrparenthesis^v & \simeq & \begin{cases} (\llparenthesis M \rrparenthesis^v, \llparenthesis N \rrparenthesis^v)_{A \times B} & \text{if } d(A \times B) \leq n \\ \langle \llparenthesis M \rrparenthesis^v, \llparenthesis N \rrparenthesis^v \rangle & \text{if } d(A \times B) = n+1 \end{cases} \\
\llparenthesis (\pi_1^{A \times B}(M)) \rrparenthesis^v & \simeq & \begin{cases} \pi_1^{A \times B}(\llparenthesis M \rrparenthesis^v) & \text{if } d(A \times B) \leq n \\ \Pi_1(\llparenthesis M \rrparenthesis^v) & \text{if } d(A \times B) = n+1 \end{cases}
\end{array}$$

$$\begin{array}{lcl}
\llparenthesis (\pi_2^{A \times B}(M)) \rrparenthesis^v & \simeq & \begin{cases} \pi_2^{A \times B}(\llparenthesis M \rrparenthesis^v) & \text{if } d(A \times B) \leq n \\ \Pi_2(\llparenthesis M \rrparenthesis^v) & \text{if } d(A \times B) = n+1 \end{cases} \\
\llparenthesis \lambda x : A.M : B \rrparenthesis^v & \simeq & \begin{cases} \lambda x : A.\llparenthesis M \rrparenthesis^{v[x:=x]} : B & \text{if } d(A \to B) \leq n \\ \langle x, \llparenthesis M \rrparenthesis^{v[x:=x]} \rangle & \text{if } d(A \to B) = n+1 \end{cases} \\
\llparenthesis M(N)_{A \to B} \rrparenthesis^v & \simeq & \begin{cases} \llparenthesis M \rrparenthesis^v(\llparenthesis N \rrparenthesis^v)_{A \to B} & \text{if } d(A \to B) \leq n \\ \llparenthesis M \rrparenthesis^v @ \llparenthesis N \rrparenthesis^v & \text{if } d(A \to B) = n+1 \end{cases} \\
\llparenthesis \{x : A \mid P\} \rrparenthesis^v & \simeq & \begin{cases} \{x : A \mid \llparenthesis P \rrparenthesis^{v[x:=x]}\} & \text{if } d(\mathrm{Set}\,(A)) \leq n \\ \langle x, \llparenthesis P \rrparenthesis^{v[x:=x]} \rangle & \text{if } d(\mathrm{Set}\,(A)) = n+1 \end{cases}
\end{array}$$

Note that this is a *partial* definition; $\llparenthesis M \rrparenthesis^v_A$ will sometimes be undefined.

Definition 5.8 (Interpretation of Small Propositions). If P is a small $\mathcal{A}_{n+1}$-proposition, we define the small proposition $\llparenthesis P \rrparenthesis^v$ of $\mathcal{A}_n$.

$$\begin{array}{lcl}
\llparenthesis M \hat{=}_{\mathbb{N}} N \rrparenthesis^v & \simeq & \llparenthesis M \rrparenthesis^v \hat{=}_{\mathbb{N}} \llparenthesis N \rrparenthesis^v \\
\llparenthesis \hat{\bot} \rrparenthesis^v & \equiv & \hat{\bot} \\
\llparenthesis P \hat{\supset} Q \rrparenthesis^v & \simeq & \llparenthesis P \rrparenthesis^v \hat{\supset} \llparenthesis Q \rrparenthesis^v \\
\llparenthesis \hat{\forall} x : \mathbb{N}.P \rrparenthesis^v & \simeq & \hat{\forall} x : \mathbb{N}.\llparenthesis P \rrparenthesis^{v[x:=x]} \\
\llparenthesis M \hat{\in}_A N \rrparenthesis^v & \simeq & \begin{cases} \llparenthesis M \rrparenthesis^v \hat{\in}_A \llparenthesis N \rrparenthesis^v & \text{if } d(A) \leq n \\ \llparenthesis N \rrparenthesis^v @ \llparenthesis M \rrparenthesis^v & \text{if } d(A) = n+1 \end{cases}
\end{array}$$

Definition 5.9 (Depth of a Proposition). We define the *depth* of a proposition ϕ, $d(\phi)$, to be

$$d(\phi) = \begin{cases} 0 & \text{if } \phi \text{ is quantifier-free} \\ \max\{d(A) \mid \phi \text{ contains a quantifier } \forall x : A\} & \text{otherwise} \end{cases}$$

Definition 5.10 (Interpretation of Propositions). If ϕ is an $\mathcal{A}_{n+1}$-proposition of depth $\leq n$, we define the $\mathcal{A}_n$-proposition $\llparenthesis \phi \rrparenthesis^v$ as follows

$$\begin{array}{lcl}
\llparenthesis M =_{\mathbb{N}} N \rrparenthesis^v & \simeq & \llparenthesis M \rrparenthesis^v =_{\mathbb{N}} \llparenthesis N \rrparenthesis^v \\
\llparenthesis \bot \rrparenthesis^v & \equiv & \bot \\
\llparenthesis \phi \supset \psi \rrparenthesis^v & \simeq & \llparenthesis \phi \rrparenthesis^v \supset \llparenthesis \psi \rrparenthesis^v \\
\llparenthesis \forall x : A . \phi \rrparenthesis^v & \simeq & \forall x : A . \llparenthesis \phi \rrparenthesis^{v[x:=x]} \\
\llparenthesis V(P) \rrparenthesis^v & \simeq & V(\llparenthesis P \rrparenthesis^v)
\end{array}$$

We have defined a sound interpretation of all the judgement forms of $\mathcal{A}_{n+1}$ except one: the judgement form $\Gamma \vdash \Phi \Rightarrow \phi$. To interpret these judgements, we shall define a notion of *satisfaction.* Intuitively, we define what it is for a proposition ϕ of $\mathcal{A}_{n+1}$ to be 'true' under a context Δ, valuation v and sequence of propositions Φ of $\mathcal{A}_n$.

Definition 5.11 (Satisfaction). Let $\Phi \equiv \phi_1, \ldots, \phi_m$ be a sequence of propositions of $\mathcal{A}_n$ such that $\Delta \vdash_n \phi_1$ Prop, $\ldots$, $\Delta \vdash_n \phi_m$ Prop. Let v be a Δ-valuation of Γ. Suppose $\Gamma \vdash \phi$ Prop. We define what it means for (Δ, Φ, v) to *satisfy* ϕ, $(\Delta, \Phi, v) \models \phi$, as follows.

If $d(\phi) \leq n$, then $((\Delta, \Phi, v) \models \phi) \Leftrightarrow (\Delta \vdash_n \Phi \Rightarrow \llparenthesis \phi \rrparenthesis^v)$.

Otherwise,

- $(\Delta, \Phi, v) \models \phi \supset \psi$ iff, for all $\Delta' \supseteq \Delta$ and $\Phi' \supseteq \Phi$, if $(\Delta', \Phi', v) \models \phi$ then $(\Delta', \Phi', v) \models \psi$.
- $(\Delta, \Phi, v) \models \forall x : A . \phi$ iff, for all $\Delta' \supseteq \Delta$ and $a \in \llbracket A \rrbracket_{\Delta'}$, we have $(\Delta', \Phi, v[x := a]) \models \phi$.

Definition 5.12 (Satisfaction and Truth). Let $\Gamma \vdash \mathcal{J}$ be a judgement of $\mathcal{A}_{n+1}$, and let v be a Δ-valuation of Γ. We define what it means for Δ and v to *satisfy* $\mathcal{J}$, written $(\Delta, v) \models \mathcal{J}$, as follows:

- $(\Delta, v) \models M : A$ iff $\llparenthesis M \rrparenthesis^v \in \llbracket A \rrbracket_\Delta$.
- $(\Delta, v) \models M = N : A$ iff $\llparenthesis M \rrparenthesis^v \sim^A_\Delta \llparenthesis N \rrparenthesis^v$.
- $(\Delta, v) \models P$ prop iff $\Delta \vdash_n \llparenthesis P \rrparenthesis^v$ prop.
- $(\Delta, v) \models P = Q$ iff $\Delta \vdash_n \llparenthesis P \rrparenthesis^v = \llparenthesis Q \rrparenthesis^v$.
- If $d(\phi) \leq n$, then $(\Delta, v) \models \phi$ Prop iff $\Delta \vdash_n \llparenthesis \phi \rrparenthesis^v$ Prop.
- $(\Delta, v) \models \phi = \psi$ iff for all Φ, $(\Delta, \Phi, v) \models \phi \Leftrightarrow (\Delta, \Phi, v) \models \psi$.
- $(\Delta, v) \models \psi_1, \ldots, \psi_n \Rightarrow \chi$ iff, for all Φ, if $(\Delta, \Phi, v) \models \psi_i$ for $1 \leq i \leq n$ then $(\Delta, \Phi, v) \models \chi$.
- For all other judgement bodies $\mathcal{J}$, we have $(\Delta, v) \models \mathcal{J}$ for all Δ, v.

We say a judgement $\Gamma \vdash \mathcal{J}$ of $\mathcal{A}_{n+1}$ is *true* iff, for every context Δ of $\mathcal{A}_n$ such that $\Delta \vdash_n$ valid and every Δ-valuation v of Γ, $(\Delta, v) \models \mathcal{J}$.

The following theorem shows that this interpretation is sound.

Theorem 5.13 (Soundness) *Every derivable judgement of $\mathcal{A}_{n+1}$ is true.*

The proof is given in Appendix Appendix B.1.

Theorem 5.14 (Completeness)

1. *Let $\Gamma \vdash \mathcal{J}$ be a judgement of $\mathcal{A}_n$, and suppose $\mathcal{J}$ does not have the form $\Phi \Rightarrow \psi$. If the judgement is true, and $\Gamma \vdash_n$* valid*, then the judgement is derivable in $\mathcal{A}_n$.*
2. *Let $\Gamma \vdash \phi_1, \ldots, \phi_m \Rightarrow \psi$ be a judgement of $\mathcal{A}_n$. If the judgement is true, and we have $\Gamma \vdash_n$* valid *and $\Gamma \vdash_n \phi_i$* Prop *for $i = 1, \ldots, m$, then the judgement is derivable in $\mathcal{A}_n$.*

Proof.

1. Let 1_Γ be the identity function on $\mathrm{dom}\,\Gamma$. Then 1_Γ is a Γ-valuation of Γ and, for every expression X of $\mathcal{A}_n$ such that $\mathrm{FV}(X) \subseteq \mathrm{dom}\,\Gamma$,
$$(\!|X|\!)^{1_\Gamma} \equiv X \ .$$
So, suppose $\Gamma \vdash M : A$ is a judgement of $\mathcal{A}_n$, and is true. Then
$$(\Gamma, 1_\Gamma) \models M : A$$
and so $\Gamma \vdash_n (\!|M|\!)^{1_\Gamma} : A$. But $(\!|M|\!)^{1_\Gamma} \equiv M$, and so $\Gamma \vdash_n M : A$ as required. The proof for the other judgement forms is similar.
2. Suppose $\Gamma \vdash \Phi \Rightarrow \psi$ is true, where $\Phi \equiv \phi_1, \ldots, \phi_m$. We have that
$$\Gamma \vdash \Phi \Rightarrow \phi_i \qquad (i = 1, \ldots, m)$$
and so $(\Gamma, \Phi, 1_\Gamma)$ satisfies each ϕ_i. Therefore, $(\Gamma, \Phi, 1_\Gamma)$ satisfies ψ, that is
$$\Gamma \vdash \Phi \Rightarrow \psi$$
as required.

Corollary 5.14.1 *If $\mathcal{J}$ is a judgement of $\mathcal{A}_n$ derivable in $\mathcal{A}_{n+1}$, then $\mathcal{J}$ is derivable in $\mathcal{A}_n$.*

Proof. This follows almost immediately from the Soundness Theorem and the Completeness Theorem. There are just two facts that need to be verified:

1. If Γ is a context of $\mathcal{A}_n$, and $\Gamma \vdash_{n+1} \mathcal{J}$, then $\Gamma \vdash_n$ valid.
2. If Γ is a context of $\mathcal{A}_n$; ϕ_1, …, ϕ_m are propositions of $\mathcal{A}_n$; and $\Gamma \vdash_{n+1} \phi_1, \ldots, \phi_m \Rightarrow \psi$; then $\Gamma \vdash_n$ valid and $\Gamma \vdash_n \phi_i$ Prop.

These are proven fairly easily by induction on derivations, using the Soundness and Completeness Theorems.

Corollary 5.14.2 (Conservativity of T_ω over T_2) *If $\mathcal{J}$ is a judgement of T_2, and $\mathcal{J}$ is derivable in T_ω, then $\mathcal{J}$ is derivable in T_2.*

Proof. Suppose $\mathcal{J}$ is derivable in T_ω. Let n be the largest depth of type or proposition that occurs in the derivation. Then $\mathcal{J}$ is derivable in $\mathcal{A}_n$. Applying Corollary 5.14.1, we have that $\mathcal{J}$ is derivable in $\mathcal{A}_{n-1}$, $\mathcal{A}_{n-2}$, $\ldots$, $\mathcal{A}_0$. But derivability in $\mathcal{A}_0$ is the same as derivability in T_2.

5.3. $\mathrm{T}_\omega U$ *is Conservative over* T_ω

The system $\mathrm{T}_\omega U$ is the fragment of LTT_0 that includes all the types of T_ω, and the universe U, but does not include types such as $U \times U$, $\mathbb{N} \to U$, or $\mathrm{Set}\,(U)$. It is defined in a similar manner to the systems $\mathcal{A}_n$ of the previous section, but using a new notion of depth.

Definition 5.15 ($\mathrm{T}_\omega U$). A type A of LTT_0 is a *type of* $\mathrm{T}_\omega U$, iff either $A \equiv U$ or the symbol U does not occur in A.

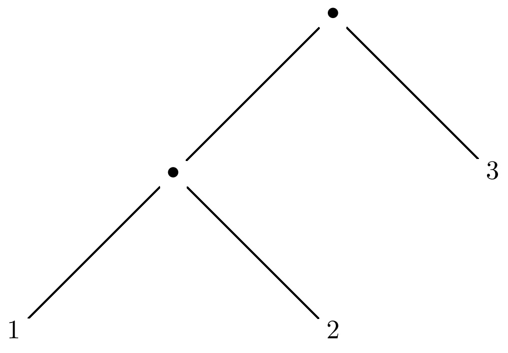
By a *term* (small proposition, proposition, context, judgement) of $\mathrm{T}_\omega U$, we mean a term (small proposition, proposition, context, judgement) of LTT_0 in which every type that occurs as a subexpression is a type of $\mathrm{T}_\omega U$

We say a judgement $\mathcal{J}$ of $\mathrm{T}_\omega U$ is *derivable* in $\mathrm{T}_\omega U$ iff there exists a derivation of $\mathcal{J}$ in LTT_0 consisting solely of judgements of $\mathrm{T}_\omega U$; that is, a derivation of $\mathcal{J}$ in which every type that occurs is a type of $\mathrm{T}_\omega U$.

We write $\Gamma \vdash^{+} \mathcal{J}$ iff the judgement $\Gamma \vdash \mathcal{J}$ is derivable in $\mathrm{T}_\omega U$, and $\Gamma \vdash^{-} \mathcal{J}$ iff the judgement $\Gamma \vdash \mathcal{J}$ is derivable in T_ω.

Note. The types of $\mathrm{T}_\omega U$ are not closed under ×, → or Set ( ). For example, the types $U \times U$ and $U \to U$ are not types of $\mathrm{T}_\omega U$.

In order to prove $\mathrm{T}_\omega U$ conservative over T_ω, we must find an interpretation of U and of the types $T(M)$. We do this by interpreting the objects of $T(M)$ as *binary trees* with leaves labelled by natural numbers. For example, the object $((1,2),3)$ of type $T((\hat{\mathbb{N}}\hat{\times}\hat{\mathbb{N}})\hat{\times}\hat{\mathbb{N}})$ will be interpreted as the binary tree



We interpret U as the set of all *shapes* of binary tree. We begin by inventing a syntax for the set of all shapes of binary trees:

Definition 5.16 (Shape). The set of *shapes* is defined inductively by:

- $\bullet$ is a shape.
- If S and T are shapes, so is $S \wedge T$.

We write $\mathscr{S}$ for the set of all shapes.

The example tree above has shape $(\bullet \wedge \bullet) \wedge \bullet$.

We must thus associate each shape with a small type. This association is done formally by the following function:

Definition 5.17. For every shape $S \in \mathscr{S}$, define the type $\mathscr{T}(S)$ of T_ω as follows:

$$\begin{array}{lcl} \mathscr{T}(\bullet) & \equiv & \mathbb{N} \\ \mathscr{T}(S \wedge T) & \equiv & \mathscr{T}(S) \times \mathscr{T}(T) \ . \end{array}$$

There are two other gaps between $\mathrm{T}_\omega U$ and T_ω to be bridged. In T_ω, we can only eliminate $\mathbb{N}$ over $\mathbb{N}$; in $\mathrm{T}_\omega U$, we can eliminate over any small type. Likewise, in T_ω, a small proposition may only involve quantification over $\mathbb{N}$; in $\mathrm{T}_\omega U$, a small proposition may involve quantification over any small type.

We bridge these gaps by using the fact that every binary tree can be *coded* as a natural number. Given a bijection $P : \mathbb{N}^2 \to \mathbb{N}$, we can assign a code number to every binary tree. The binary tree above, for example, would be assigned the code number $P(P(1,2),3)$. We shall define, for every shape S, mutually inverse functions

$$\begin{array}{lcl} \mathrm{code}_S & : & \mathscr{T}(S) \to \mathbb{N} \\ \mathrm{decode}_S & : & \mathbb{N} \to \mathscr{T}(S) \end{array}$$

Using these functions, we can interpret recursion over small types by recursion over $\mathbb{N}$, and quantification over small types by quantification over $\mathbb{N}$.

We turn now to the formal details. The first step is to construct in T_ω the bijection P above, and the coding and decoding functions.

Lemma 5.18 (Pairing Function) *There exist* T_ω*-terms*

$$\begin{array}{lcl} \mathbf{P} & : & \mathbb{N} \times \mathbb{N} \to \mathbb{N} \\ \mathbf{Q}_1 & : & \mathbb{N} \to \mathbb{N} \\ \mathbf{Q}_2 & : & \mathbb{N} \to \mathbb{N} \end{array}$$

such that the following are theorems of T_ω*:*

$$\left.\begin{array}{c} \forall x : \mathbb{N}. \forall y : \mathbb{N}. \mathbf{Q}_1(\mathbf{P}(x,y)) =_{\mathbb{N}} x \\ \forall x : \mathbb{N}. \forall y : \mathbb{N}. \mathbf{Q}_2(\mathbf{P}(x,y)) =_{\mathbb{N}} y \\ \forall x : \mathbb{N}. x =_{\mathbb{N}} \mathbf{P}(\mathbf{Q}_1(x), \mathbf{Q}_2(x)) \end{array}\right\} \qquad (9)$$

PROOF. Consider the three primitive recursive functions

$$\begin{array}{lcl} p(m,n) & = & 2^m(2n+1) \\ q(n) & = & \text{the greatest } m \text{ such that } 2^m \text{ divides } n \\ r(n) & = & 1/2(n/2^{q(n)} - 1) \end{array}$$

It is straightforward to define terms $\mathbf{P}$, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ in T_ω that express p, q and r and prove the three formulas (9).

Fix three such terms $\mathbf{P}$, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ for the sequel.

We shall also need a notion of equality on every small type in T_ω, not just $\mathbb{N}$. This is defined as follows.

Definition 5.19. Given T_ω-terms M and N and a T_ω-type A, define the T_ω-proposition $M =_A N$ as follows.

$$\begin{array}{lcl} M =_{\mathbb{N}} N & \equiv & M =_{\mathbb{N}} N \\ M =_{A \times B} N & \equiv & \pi_1(M) =_A \pi_1(N) \wedge \pi_2(M) =_B \pi_2(N) \\ M =_{A \to B} N & \equiv & \forall x : A.M(x) =_B N(x) \\ M =_{\mathrm{Set}(A)} N & \equiv & \forall x : A.(x \in_A M \leftrightarrow x \in_A N) \end{array}$$

Definition 5.20 (Coding Functions). For each shape $S \in \mathscr{S}$, define the T_ω-terms

$$\begin{array}{lcl} \mathrm{code}_S & : & \mathscr{T}(S) \to \mathbb{N} \\ \mathrm{decode}_S & : & \mathbb{N} \to \mathscr{T}(S) \end{array}$$

as follows.

$$\begin{array}{lcl} \mathrm{code}_\bullet & \equiv & \lambda x : \mathbb{N}.x \\ \mathrm{decode}_\bullet & \equiv & \lambda x : \mathbb{N}.x \\ & & \\ \mathrm{code}_{S \wedge T} & \equiv & \lambda p : \mathscr{T}(S) \times \mathscr{T}(T).\mathbf{P}(\mathrm{code}_S(\pi_1(p)), \mathrm{code}_T(\pi_2(p))) \\ \mathrm{decode}_{S \wedge T} & \equiv & \lambda n : \mathbb{N}.(\mathrm{decode}_S(\mathbf{Q}_1(n)), \mathrm{decode}_T(\mathbf{Q}_2(n))) \end{array}$$

Lemma 5.21 *For every shape S, the following are theorems of T_ω:*

$$\forall p : \mathscr{T}(S).\mathrm{decode}_S(\mathrm{code}_S(p)) =_{\mathscr{T}(S)} p$$
$$\forall n : \mathbb{N}.\mathrm{code}_S(\mathrm{decode}_S(n)) =_{\mathbb{N}} n$$

PROOF. The proof is by induction on S, using the properties of $\mathbf{P}$, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ from Lemma 5.18.

We can now proceed to define our interpretation of $\mathrm{T}_\omega U$ in terms of T_ω. The definition is more complex than the interpretation in the previous section, because the type-theoretic component of $\mathrm{T}_\omega U$ is dependent, so we must define our interpretations of terms and types simultaneously.

Definition 5.22. Let Δ be a context of T_ω, and v a function. We define the following simultaneously.

- Given a $\mathrm{T}_\omega U$-term M and a function v, define the object $\llparenthesis M \rrparenthesis^v$ as follows.

$$\begin{array}{lcl}
\llparenthesis x \rrparenthesis^v & \simeq & v(x) \\
\llparenthesis 0 \rrparenthesis^v & \equiv & 0 \\
\llparenthesis \mathrm{s}\, M \rrparenthesis^v & \simeq & \mathrm{s}\llparenthesis M \rrparenthesis^v \\
\llparenthesis (M, N)_{A\times B} \rrparenthesis^v & \simeq & (\llparenthesis M \rrparenthesis^v, \llparenthesis N \rrparenthesis^v)_{\llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v} \\
\llparenthesis \pi_1^{A\times B}(M) \rrparenthesis^v & \simeq & \pi_1^{\llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v}(\llparenthesis M \rrparenthesis^v) \\
\llparenthesis \pi_2^{A\times B}(M) \rrparenthesis^v & \simeq & \pi_2^{\llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v}(\llparenthesis M \rrparenthesis^v) \\
\llparenthesis \lambda x : A.M : B \rrparenthesis^v & \simeq & \lambda x : \llparenthesis A \rrparenthesis^v . \llparenthesis M \rrparenthesis^{v[x:=x]} : \llparenthesis B \rrparenthesis^v \\
\llparenthesis M(N)_{A\to B} \rrparenthesis^v & \simeq & \llparenthesis M \rrparenthesis^v(\llparenthesis N \rrparenthesis^v)_{\llparenthesis A \rrparenthesis^v \to \llparenthesis B \rrparenthesis^v} \\
\llparenthesis \hat{\mathbb{N}} \rrparenthesis^v & = & \bullet \\
\llparenthesis M \hat{\times} N \rrparenthesis^v & \simeq & \llparenthesis M \rrparenthesis^v \wedge \llparenthesis N \rrparenthesis^v \\
\llparenthesis \{x : A \mid P\} \rrparenthesis^v & \simeq & \{x : \llparenthesis A \rrparenthesis^v \mid \llparenthesis P \rrparenthesis^{v[x:=x]}\} \\
\llparenthesis \mathrm{E}_{\mathbb{N}}([x]T(K), L, [x,y]M, N) \rrparenthesis^v & \simeq & \mathrm{decode}_{S(\llparenthesis N \rrparenthesis^v)}(\mathrm{R}(\mathrm{code}_{S(0)}(\llparenthesis L \rrparenthesis^v), \\
 & & \quad [x,y]\mathrm{code}_{S(\mathrm{s}\, x)}(\llparenthesis M \rrparenthesis^{v'}), \llparenthesis N \rrparenthesis^v))
\end{array}$$

  where $S(N) \equiv \llparenthesis K \rrparenthesis^{v[x:=N]}$ and $v' = v[x := x, y := \mathrm{decode}_{S(x)}(y)]$.

- Given a type $A \not\equiv U$ of $\mathrm{T}_\omega U$, define a type $\llparenthesis A \rrparenthesis^v$ of T_ω.

$$\begin{array}{lcl}
\llparenthesis \mathbb{N} \rrparenthesis^v & \equiv & \mathbb{N} \\
\llparenthesis A \times B \rrparenthesis^v & \simeq & \llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v \\
\llparenthesis A \to B \rrparenthesis^v & \simeq & \llparenthesis A \rrparenthesis^v \to \llparenthesis B \rrparenthesis^v \\
\llparenthesis T(M) \rrparenthesis^v & \simeq & T(\llparenthesis M \rrparenthesis^v) \\
\llparenthesis \mathrm{Set}\,(A) \rrparenthesis^v & \simeq & \mathrm{Set}\,(\llparenthesis A \rrparenthesis^v)
\end{array}$$

- Given a $\mathrm{T}_\omega U$-type A, define a set $\llbracket A \rrbracket^v_\Delta$ and an equivalence relation $\sim^A_{\Delta v}$ on $\llbracket A \rrbracket^v_\Delta$ as follows.

  If $A \not\equiv U$, then

$$\begin{array}{lcl}
\llbracket A \rrbracket^v & = & \{M \mid \Delta \vdash^+ M : \llparenthesis A \rrparenthesis^v\} \\
M \sim^A_{\Delta v} N & \Leftrightarrow & \Delta \vdash^+ \Rightarrow M =_{\llparenthesis A \rrparenthesis^v} N
\end{array}$$

  Otherwise,

$$\begin{array}{lcl}
\llbracket U \rrbracket^v & = & \mathscr{S} \\
S \sim^U_{\Delta v} T & \Leftrightarrow & S = T
\end{array}$$

- Let $\Gamma \equiv x_1 : A_1, \ldots, x_m : A_m$ be a context of $\mathrm{T}_\omega U$. We say that v is a Δ-*valuation* of Γ iff $v(x_i) \in \llbracket A_i \rrbracket^v_\Delta$ for $i = 1, \ldots, n$.

- Given a small proposition P of $\mathrm{T}_\omega U$, define a small proposition $\llparenthesis P \rrparenthesis^v$ of T_ω as follows.

$$\begin{array}{lcl}
\llparenthesis M_1 \hat{=}_N M_2 \rrparenthesis^v & \simeq & \mathrm{code}_{\llparenthesis N \rrparenthesis^v}(\llparenthesis M_1 \rrparenthesis^v) \hat{=}_{\mathbb{N}} \mathrm{code}_{\llparenthesis N \rrparenthesis^v}(\llparenthesis M_2 \rrparenthesis^v) \\
\llparenthesis \hat{\bot} \rrparenthesis^v & \equiv & \hat{\bot} \\
\llparenthesis P \hat{\supset} Q \rrparenthesis^v & \equiv & \llparenthesis P \rrparenthesis^v \hat{\supset} \llparenthesis Q \rrparenthesis^v \\
\llparenthesis \hat{\forall} x : M . P \rrparenthesis^v & \simeq & \hat{\forall} x : \mathbb{N} . \llparenthesis P \rrparenthesis^{v[x := \mathrm{decode}_{\llparenthesis M \rrparenthesis^v}(x)]} \\
\llparenthesis M \hat{\in}_A N \rrparenthesis^v & \simeq & \llparenthesis M \rrparenthesis^v \hat{\in}_{\llparenthesis A \rrparenthesis^v} \llparenthesis N \rrparenthesis^v
\end{array}$$

- Given a proposition ϕ of $\mathrm{T}_\omega U$ that does not include a quantifier over U, define a proposition $\llparenthesis \phi \rrparenthesis^v$ of T_ω as follows.

$$\begin{array}{lcl}
\llparenthesis M_1 =_N M_2 \rrparenthesis^v & \simeq & \llparenthesis M_1 \rrparenthesis^v =_{\mathscr{T}(\llparenthesis N \rrparenthesis^v)} \llparenthesis M_2 \rrparenthesis^v \\
\llparenthesis \bot \rrparenthesis^v & \equiv & \bot \\
\llparenthesis \phi \supset \psi \rrparenthesis^v & \simeq & \llparenthesis \phi \rrparenthesis^v \supset \llparenthesis \psi \rrparenthesis^v \\
\llparenthesis \forall x : A . \phi \rrparenthesis^v & \simeq & \forall x : \llparenthesis A \rrparenthesis^v . \llparenthesis \phi \rrparenthesis^{v[x:=x]} \\
\llparenthesis V(P) \rrparenthesis^v & \simeq & V(\llparenthesis P \rrparenthesis^v)
\end{array}$$

Recall that we write $\Delta \vdash^- \mathcal{J}$ iff $\Delta \vdash \mathcal{J}$ is derivable in T_ω.

Definition 5.23 (Satisfaction). Let $\Phi \equiv \phi_1, \ldots, \phi_m$ be a sequence of propositions of T_ω such that $\Delta \vdash^- \phi_i$ Prop. Let v be a Δ-valuation of Γ. Suppose $\Gamma \vdash \phi$ Prop. We define what it means for (Δ, Φ, v) to *satisfy* ϕ, $(\Delta, \Phi, v) \models \phi$, as follows.

If ϕ does not involve quantification over U, then

$$((\Delta, \Phi, v) \models \phi) \Leftrightarrow (\Delta \vdash^- \Phi \Rightarrow \llparenthesis \phi \rrparenthesis^v) \enspace .$$

Otherwise,

- $(\Delta, \Phi, v) \models \phi \supset \psi$ iff, for all $\Delta' \supseteq \Delta$ and $\Phi' \supseteq \Phi$, if $(\Delta', \Phi', v) \models \phi$ then $(\Delta', \Phi', v) \models \psi$.
- $(\Delta, \Phi, v) \models \forall x : A . \phi$ iff, for all $\Delta' \supseteq \Delta$ and $a \in [\![A]\!]^v_\Delta$, we have $(\Delta', \Phi, v[x := a]) \models \phi$.

Definition 5.24 (Satisfaction and Truth). Let $\Gamma \vdash \mathcal{J}$ be a judgement of $\mathrm{T}_\omega U$. Let $\Delta \vdash^-$ valid, and let v be a Δ-valuation of Γ. We define what it means for Δ and v to *satisfy* $\mathcal{J}$, $(\Delta, v) \models \mathcal{J}$, as follows.

- If $A \not\equiv U$, then $(\Delta, v) \models A$ type iff $\llparenthesis A \rrparenthesis^v$ is defined.
- If $A \not\equiv U \not\equiv B$, then $(\Delta, v) \models A = B$ iff $\llparenthesis A \rrparenthesis^v \equiv \llparenthesis B \rrparenthesis^v$.
- $(\Delta, v) \models M : A$ iff $\llparenthesis M \rrparenthesis^v \in [\![A]\!]^v_\Delta$.
- $(\Delta, v) \models M = N : A$ iff $\llparenthesis M \rrparenthesis^v \sim^A_{\Delta v} \llparenthesis N \rrparenthesis^v$.

- $(\Delta, v) \models P$ prop iff $\Delta \vdash^- (\!|P|\!)^v$ prop.
- $(\Delta, v) \models P = Q$ iff $\Delta \vdash^- \Rightarrow V((\!|P|\!)^v) \leftrightarrow V((\!|Q|\!)^v)$.
- If ϕ does not include a quantifier over U, then $(\Delta, v) \models \phi$ Prop iff $\Delta \vdash^- (\!|\phi|\!)$ Prop.
- $(\Delta, v) \models \phi = \psi$ iff, for all Φ, we have $(\Delta, \Phi, v) \models \phi$ iff $(\Delta, \Phi, v) \models \psi$.
- $(\Delta, v) \models \phi_1, \ldots, \phi_m \Rightarrow \psi$ iff, for all Φ, if $(\Delta, \Phi, v) \models \phi_i$ for $i = 1, \ldots, m$ then $(\Delta, \Phi, v) \models \psi$.
- For all other judgement forms, we have $(\Delta, v) \models \mathcal{J}$ for all Δ, v.

We say a judgement $\Gamma \vdash \mathcal{J}$ of $\mathrm{T}_\omega U$ is *true* iff, for all Δ such that $\Delta \vdash^-$ valid and all Δ-valuations v of Γ, $(\Delta, v) \models \mathcal{J}$.

Remark. This interpretation uses the propositional equality defined in Definition 5.19, whereas our interpretation in the previous section used judgemental equality. This is because the properties of our coding and decoding functions can be shown to hold up to propositional equality (as in Lemma 5.21), but not up to judgemental equality.

We now prove that the interpretation is sound.

Theorem 5.25 (Soundness) *Every derivable judgement in* $\mathrm{T}_\omega U$ *is true.*

The proof is given in Appendix Appendix B.2.

Theorem 5.26 (Completeness) *If* $\Gamma \vdash \mathcal{J}$ *is a judgement of* T_ω *that is true, and* $\Gamma \vdash^-$ valid*, then* $\Gamma \vdash \mathcal{J}$ *is derivable in* T_ω*.*

Proof. Exactly as in Theorem 5.14.

Corollary 5.26.1 *If* $\mathcal{J}$ *is a judgement of* T_ω *derivable in* $\mathrm{T}_\omega U$*, then* $\mathcal{J}$ *is derivable in* T_ω*.*

Proof. Similar to Corollary 5.14.1.

5.4. LTT$_0$ *is Conservative over* $\mathrm{T}_\omega U$

The next step in our proof is to apply the same method to show that LTT$_0$ is conservative over $\mathrm{T}_\omega U$. The proof is very similar to Section 5.2, but the details are more complicated, because we are now dealing with LTTs whose type theoretic components use dependent types.

Once again, we introduce an infinite sequence of subsystems between $\mathrm{T}_\omega U$ and LTT$_0$:

$$\mathrm{T}_\omega U = \mathcal{B}_0 \hookrightarrow \mathcal{B}_1 \hookrightarrow \mathcal{B}_2 \hookrightarrow \cdots \mathrm{LTT}_0$$

We do this using a new definition of the depth of a type:

Definition 5.27 (Depth). Define the *depth* $D(A)$ of a type A of LTT_0 by

$$\begin{array}{lcl}
D(\mathbb{N}) & = & 0 \\
D(A \times B) & = & \begin{cases} 0 & \text{if } D(A) = D(B) = 0 \\ \max(D(A), D(B)) + 1 & \text{otherwise} \end{cases} \\
D(A \to B) & = & \begin{cases} 0 & \text{if } D(A) = D(B) = 0 \\ \max(D(A), D(B)) + 1 & \text{otherwise} \end{cases} \\
D(\mathrm{Set}\,(A)) & = & \begin{cases} 0 & \text{if } D(A) = 0 \\ D(A) + 1 & \text{otherwise} \end{cases} \\
D(U) & = & 1 \\
D(T(M)) & = & 0
\end{array}$$

We define the depth of a proposition ϕ, $D(\phi)$, to be the largest depth of a type A such that the quantifier $\forall x : A$ occurs in ϕ, or $D(\phi) = 0$ if ϕ is quantifier-free.

Note that the types of $\mathrm{T}_\omega U$ are exactly the types A such that $D(A) \leq 1$.

The subsystems $\mathcal{B}_n$ are defined as follows.

Definition 5.28 ($\mathcal{B}_n$). Let $n \geq 0$. By a *type* (term, small proposition, proposition, context, judgement) *of* $\mathcal{B}_n$, we mean a type (term, small proposition, proposition, context, judgement) of LTT_0 that does not contain, as a subexpression, any type A such that $D(A) > n$.

We say a judgement $\mathcal{J}$ of $\mathcal{B}_n$ is *derivable* in $\mathcal{B}_n$ iff there exists a derivation of $\mathcal{J}$ in LTT_0 consisting solely of judgements of $\mathcal{B}_n$; that is, a derivation of $\mathcal{J}$ in which no type A occurs such that $D(A) > n$. In this section, we write $\Gamma \vdash_n \mathcal{J}$ iff the judgement $\Gamma \vdash \mathcal{J}$ is derivable in $\mathcal{B}_n$.

We define an interpretation of $\mathcal{B}_{n+1}$ in terms of $\mathcal{B}_n$:

Definition 5.29. Fix $n \geq 1$. Let Δ be a context of $\mathcal{B}_n$, and v a function. We define the following simultaneously.

- Given a term M of $\mathcal{B}_{n+1}$, define the object $\llparenthesis B \rrparenthesis^v$.

$$
\begin{array}{lll}
\llparenthesis x \rrparenthesis^v & \simeq & v(x) \\
\llparenthesis 0 \rrparenthesis^v & \equiv & 0 \\
\llparenthesis \mathrm{s}\, M \rrparenthesis^v & \simeq & \mathrm{s}\llparenthesis M \rrparenthesis^v \\
\multicolumn{3}{l}{\llparenthesis \mathrm{E}_{\mathbb{N}}([x]T(K), L, [x,y]M, N) \rrparenthesis^v} \\
 & \simeq & \mathrm{E}_{\mathbb{N}}([x]T(\llparenthesis K \rrparenthesis^{v[x:=x]}), \llparenthesis L \rrparenthesis^v, \\
 & & \quad [x,y]\llparenthesis M \rrparenthesis^{v[x:=x,y:=y]}, \llparenthesis N \rrparenthesis^v) \\
\llparenthesis (M,N)_{A\times B} \rrparenthesis^v & \simeq & \begin{cases} (\llparenthesis M \rrparenthesis^v, \llparenthesis N \rrparenthesis^v)_{\llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v} & \text{if } D(A\times B) \leq n \\ \langle \llparenthesis M \rrparenthesis^v, \llparenthesis N \rrparenthesis^v \rangle & \text{if } D(A\times B) = n+1 \end{cases} \\
\llparenthesis \pi_1^{A\times B}(M) \rrparenthesis^v & \simeq & \begin{cases} \pi_1^{\llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v}(\llparenthesis M \rrparenthesis^v) & \text{if } D(A\times B) \leq n \\ \Pi_1(\llparenthesis M \rrparenthesis^v) & \text{if } D(A\times B) = n+1 \end{cases} \\
\llparenthesis \pi_2^{A\times B}(M) \rrparenthesis^v & \simeq & \begin{cases} \pi_2^{\llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v}(\llparenthesis M \rrparenthesis^v) & \text{if } D(A\times B) \leq n \\ \Pi_2(\llparenthesis M \rrparenthesis^v) & \text{if } D(A\times B) = n+1 \end{cases} \\
\llparenthesis \lambda x : A.M : B \rrparenthesis^v & \simeq & \begin{cases} \lambda x : \llparenthesis A \rrparenthesis^v.\llparenthesis M \rrparenthesis^{v[x:=x]} : \llparenthesis B \rrparenthesis^v & \\ & \text{if } D(A\to B) \leq n \\ \langle x, \llparenthesis M \rrparenthesis^{v[x:=x]} \rangle & \text{if } D(A\to B) = n+1 \end{cases} \\
\llparenthesis M(N)_{A\to B} \rrparenthesis^v & \simeq & \begin{cases} \llparenthesis M \rrparenthesis^v(\llparenthesis N \rrparenthesis^v)_{\llparenthesis A \rrparenthesis^v \to \llparenthesis B \rrparenthesis^v} & \text{if } D(A\to B) \leq n \\ \llparenthesis M \rrparenthesis^v @ \llparenthesis N \rrparenthesis^v & \text{if } D(A\to B) = n+1 \end{cases} \\
\llparenthesis \hat{\mathbb{N}} \rrparenthesis^v & \equiv & \hat{\mathbb{N}} \\
\llparenthesis M \hat{\times} N \rrparenthesis^v & \simeq & \llparenthesis M \rrparenthesis^v \hat{\times} \llparenthesis N \rrparenthesis^v \\
\llparenthesis \{x : A \mid P\} \rrparenthesis^v & \simeq & \begin{cases} \{x : \llparenthesis A \rrparenthesis^v \mid \llparenthesis P \rrparenthesis^{v[x:=x]}\} & \text{if } D(\mathrm{Set}\,(A)) \leq n \\ \langle x, \llparenthesis P \rrparenthesis^{v[x:=x]} \rangle & \text{if } D(\mathrm{Set}\,(A)) = n+1 \end{cases}
\end{array}
$$

- Given a type A of $\mathcal{B}_{n+1}$ such that $D(A) \leq n$, define the type $\llparenthesis A \rrparenthesis^v$ of $\mathcal{B}_n$.

$$
\begin{array}{lll}
\llparenthesis \mathbb{N} \rrparenthesis^v & \equiv & \mathbb{N} \\
\llparenthesis A \times B \rrparenthesis^v & \simeq & \llparenthesis A \rrparenthesis^v \times \llparenthesis B \rrparenthesis^v \\
\llparenthesis A \to B \rrparenthesis^v & \simeq & \llparenthesis A \rrparenthesis^v \to \llparenthesis B \rrparenthesis^v \\
\llparenthesis U \rrparenthesis^v & \equiv & U \\
\llparenthesis T(M) \rrparenthesis^v & \simeq & T(\llparenthesis M \rrparenthesis^v) \\
\llparenthesis \mathrm{Set}\,(A) \rrparenthesis^v & \simeq & \mathrm{Set}\,(\llparenthesis A \rrparenthesis^v)
\end{array}
$$

- Given a small proposition P of $\mathcal{B}_{n+1}$, define the small proposition $\llparenthesis P \rrparenthesis^v$

as follows.

$$\begin{array}{lcl}
(\!|M_1 \hat{=}_N M_2|\!)^v & \simeq & (\!|M_1|\!)^v \hat{=}_{(\!|N|\!)^v} (\!|M_2|\!)^v \\
(\!|\hat{\bot}|\!)^v & \equiv & \hat{\bot} \\
(\!|P \hat{\supset} Q|\!)^v & \simeq & (\!|P|\!)^v \hat{\supset} (\!|Q|\!)^v \\
(\!|\hat{\forall} x : M.P|\!)^v & \simeq & \hat{\forall} x : (\!|M|\!)^v.(\!|P|\!)^{v[x:=x]} \\
(\!|M \hat{\in}_A N|\!)^v & \simeq & (\!|M|\!)^v \hat{\in}_{(\!|A|\!)^v} (\!|N|\!)^v
\end{array}$$

- Given a type A of $\mathcal{B}_{n+1}$, define a set $[\![A]\!]^v_\Delta$ and an equivalence relation $\sim^A_{\Delta v}$ on this set.

  If $D(A) \leq n$, then

  $$\begin{array}{lcl}
  [\![A]\!]^v_\Delta & = & \{M \mid \Delta \vdash_n M : (\!|A|\!)^v\} \\
  M \sim^A_{\Delta v} N & \Leftrightarrow & \Delta \vdash_n M = N : (\!|A|\!)^v
  \end{array}$$

  Otherwise,

  $$\begin{array}{lcl}
  [\![A \times B]\!]^v_\Delta & = & \{\langle M, N\rangle \mid \Delta \vdash_n M : (\!|A|\!)^v, \Delta \vdash_n N : (\!|B|\!)^v\} \\
  \langle M, N\rangle \sim^{A\times B}_{\Delta v} \langle M', N'\rangle & \Leftrightarrow & \Delta \vdash_n M = M' : (\!|A|\!)^v \\
  & & \wedge \Delta \vdash_n N = N' : (\!|B|\!)^v \\
  & & \\
  [\![A \to B]\!]^v_\Delta & = & \{\langle x, M\rangle \mid \Delta, x : (\!|A|\!)^v \vdash M : (\!|B|\!)^v\} \\
  \langle x, M\rangle \sim^{A\to B}_{\Delta v} \langle x, M'\rangle & \Leftrightarrow & \Delta, x : (\!|A|\!)^v \vdash M = M' : (\!|B|\!)^v \\
  & & \\
  [\![\mathrm{Set}\,(A)]\!]^v_\Delta & = & \{\langle x, P\rangle \mid \Delta, x : (\!|A|\!)^v \vdash P \text{ prop}\} \\
  \langle x, P\rangle \sim^{\mathrm{Set}(A)}_{\Delta v} \langle x, P'\rangle & \Leftrightarrow & \Delta, x : (\!|A|\!)^v \vdash P = P'
  \end{array}$$

- Given a context $\Gamma \equiv x_1 : A_1, \ldots, x_m : A_m$ of $\mathcal{B}_{n+1}$, we say that v is a Δ-*valuation* of Γ iff $v(x_i) \in [\![A_i]\!]^v_\Delta$ for each i.

- Given a proposition ϕ of $\mathcal{B}_{n+1}$ such that $D(\phi) \leq n$, define the proposition $(\!|\phi|\!)^v$ as follows.

  $$\begin{array}{lcl}
  (\!|M_1 =_N M_2|\!)^v & \simeq & (\!|M_1|\!)^v =_{(\!|N|\!)^v} (\!|M_2|\!)^v \\
  (\!|\bot|\!)^v & \equiv & \bot \\
  (\!|\phi \supset \psi|\!)^v & \simeq & (\!|\phi|\!)^v \supset (\!|\psi|\!)^v \\
  (\!|\forall x : A.\phi|\!)^v & \simeq & \forall x : (\!|A|\!)^v.(\!|\phi|\!)^{v[x:=x]} \\
  (\!|V(P)|\!)^v & \simeq & V((\!|P|\!)^v)
  \end{array}$$

We define what the notion of *satisfaction* $(\Delta, \Phi, v) \models \psi$ similarly to Definition 5.11:

Definition 5.30 (Satisfaction). Let $\Phi \equiv \phi_1, \ldots, \phi_m$ be a sequence of propositions of $\mathcal{A}_n$ such that $\Delta \vdash_n \phi_1$ Prop, …, $\Delta \vdash_n \phi_m$ Prop. Let v be a Δ-valuation of Γ. Suppose $\Gamma \vdash_{n+1} \phi$ Prop. We define what it means for (Δ, Φ, v) to *satisfy* ϕ, $(\Delta, \Phi, v) \models \phi$, as follows.

If $D(\phi) \leq n$, then $((\Delta, \Phi, v) \models \phi) \Leftrightarrow (\Delta \vdash_n \Phi \Rightarrow \llparenthesis \phi \rrparenthesis^v)$.

Otherwise,

- $(\Delta, \Phi, v) \models \phi \supset \psi$ iff, for all $\Delta' \supseteq \Delta$ and $\Phi' \supseteq \Phi$, if $(\Delta', \Phi', v) \models \phi$ then $(\Delta', \Phi', v) \models \psi$.
- $(\Delta, \Phi, v) \models \forall x : A.\phi$ iff, for all $\Delta' \supseteq \Delta$ and $a \in \llbracket A \rrbracket^v_{\Delta'}$, we have $(\Delta', \Phi, v[x := a]) \models \phi$.

Definition 5.31 (Satisfaction and Truth). Let $\Gamma \vdash \mathcal{J}$ be a judgement of $\mathcal{B}_{n+1}$. Let $\Delta \vdash_n$ valid and v be a Δ-valuation of Γ. We define what it means for Δ and v to *satisfy* $\mathcal{J}$, $(\Delta, v) \models \mathcal{J}$, as follows.

- If $D(A) \leq n$, then $(\Delta, v) \models A$ type iff $\llbracket A \rrbracket^v_\Delta$ is defined and $\Delta \vdash_n \llparenthesis A \rrparenthesis^v$ type.

  If $D(A) = n + 1$, then $(\Delta, v) \models A$ type iff $\llbracket A \rrbracket^v_\Delta$ is defined.
- If $D(A), D(B) \leq n$, then $(\Delta, v) \models A = B$ iff $\llbracket A \rrbracket^v_\Delta = \llbracket B \rrbracket^v_\Delta$ and $(\sim^A_{\Delta v}) = (\sim^B_{\Delta v})$ and $\Delta \vdash_n \llparenthesis A \rrparenthesis^v = \llparenthesis B \rrparenthesis^v$.

  If $D(A) = D(B) = n + 1$, then $(\Delta, v) \models A = B$ iff $\llbracket A \rrbracket^v_\Delta = \llbracket B \rrbracket^v_\Delta$ and $(\sim^A_{\Delta v}) = (\sim^B_{\Delta v})$.
- $(\Delta, v) \models M : A$ iff $\llparenthesis M \rrparenthesis^v \in \llbracket A \rrbracket^v_\Delta$.
- $(\Delta, v) \models M = N : A$ iff $\llparenthesis M \rrparenthesis^v \sim^A_{\Delta v} \llparenthesis N \rrparenthesis^v$
- $(\Delta, v) \models P$ prop iff $\Delta \vdash_n \llparenthesis P \rrparenthesis^v$ prop.
- $(\Delta, v) \models P = Q$ iff $\Delta \vdash_n \llparenthesis P \rrparenthesis^v = \llparenthesis Q \rrparenthesis^v$
- If $D(\phi) \leq n$, then $(\Delta, v) \models \phi$ Prop iff $\Delta \vdash_n \llparenthesis \phi \rrparenthesis^v$ Prop.
- $(\Delta, v) \models \phi = \psi$ iff, for all Φ, we have $(\Delta, \Phi, v) \models \phi$ iff $(\Delta, \Phi, v) \models \psi$.
- $(\Delta, v) \models \psi_1, \ldots, \psi_m \Rightarrow \chi$ iff, for all Φ, if (Δ, Φ, v) satisfies ψ_i for all i, then (Δ, Φ, v) satisfies χ.
- For any other $\mathcal{J}$, we have $(\Delta, v) \models \mathcal{J}$ for all Δ, v.

We say $\Gamma \vdash \mathcal{J}$ is *true* iff, whenever $\Delta \vdash_n$ valid and v is a Δ-valuation of Γ, then $(\Delta, v) \models \mathcal{J}$.

Theorem 5.32 (Soundness) *Every derivable judgement in $\mathcal{B}_{n+1}$ is true.*

Proof. Similar to Theorems 5.13 and 5.25.

Theorem 5.33

1. *Let $\Gamma \vdash \mathcal{J}$ be a judgement of $\mathcal{B}_n$, and suppose $\mathcal{J}$ does not have the form $\Phi \Rightarrow \psi$. If the judgement is true, and $\Gamma \vdash_n$* valid*, then the judgement is derivable in $\mathcal{B}_n$.*
2. *Let $\Gamma \vdash \phi_1, \ldots, \phi_m \Rightarrow \psi$ be a judgement of $\mathcal{B}_n$. If the judgement is true, and we have $\Gamma \vdash_n$* valid *and $\Gamma \vdash_n \phi_i$* Prop *for $i = 1, \ldots, m$, then the judgement is derivable in $\mathcal{B}_n$.*

Proof. Similar to Theorem 5.14.

Corollary 5.33.1 *If $\mathcal{J}$ is a judgement of $\mathcal{B}_n$ derivable in $\mathcal{B}_{n+1}$, then $\mathcal{J}$ is derivable in $\mathcal{B}_n$.*

Corollary 5.33.2 *If $\mathcal{J}$ is a judgement of $\mathrm{T}_\omega U$ derivable in LTT_0, then $\mathcal{J}$ is derivable in $\mathrm{T}_\omega U$.*

With this final step, we have now completed the proof of the conservativity of LTT_0 over ACA_0:

Corollary 5.33.3 *Let ϕ be a formula of second order arithmetic with free variables $x_1, \ldots, x_m, X_1, \ldots, X_n$. If*

$$x_1 : \mathbb{N}, \ldots, x_m : \mathbb{N}, X_1 : \mathrm{Set}\,(\mathbb{N}), \ldots, X_n : \mathrm{Set}\,(\mathbb{N}) \vdash \Rightarrow \langle\!|\phi|\!\rangle$$

in LTT_0 then $\mathrm{ACA}_0 \vdash \phi$.

Proof. Let $\mathcal{J}$ be the judgement
$x_1 : \mathbb{N}, \ldots, x_m : \mathbb{N}, X_1 : \mathrm{Set}\,(\mathbb{N}), \ldots, X_n : \mathrm{Set}\,(\mathbb{N}) \vdash \Rightarrow \langle\!|\phi|\!\rangle$.

Suppose $\mathcal{J}$ is derivable in LTT_0. Then

$$\begin{array}{ll} \mathcal{J} \text{ is derivable in } \mathrm{T}_\omega U & \text{(Corollary 5.33.2)} \\ \therefore \mathcal{J} \text{ is derivable in } \mathrm{T}_\omega & \text{(Corollary 5.26.1)} \\ \therefore \mathcal{J} \text{ is derivable in } \mathrm{T}_2 & \text{(Corollary 5.14.2)} \\ \therefore \mathrm{ACA}_0 \vdash \phi & \text{(Corollary 4.3.1)} \end{array}$$

6. Other Conservativity Results

6.1. Conservativity of LTT_0^* *over ACA*

Our proof method can be adapted quite straightforwardly to prove the conservativity of LTT_0^* over ACA. We shall present these proofs briefly, giving only the details that need to be changed.

We define subsystems of LTT_0^*:

$$\mathrm{T}_2^* \hookrightarrow \mathrm{T}_\omega^* \hookrightarrow \mathrm{T}_\omega U^* \hookrightarrow \mathrm{LTT}_0^*$$

T_2^* is formed from T_2 by allowing the rule $(\mathrm{Ind}_\mathbb{N})$ to be applied with any analytic proposition ϕ. In the same manner, T_ω^* is formed from T_ω, $\mathrm{T}_\omega U^*$ is formed from $\mathrm{T}_\omega U$, and LTT_0^* is formed from LTT_0.

The proof of the conservativity of LTT_0^* over T_2^* follows exactly the same pattern as in Section 5.

Theorem 6.1 *Theorem 4.2 holds for* T_2^* *and* ACA.

Proof. Similar to the proof of Theorem 4.2.

Similarly, Corollary 5.14.2 holds for T_ω^* and T_2^*, Corollary 5.26.1 holds for $\mathrm{T}_\omega U^*$ and T_ω^*, and Corollary 5.33.2 holds for LTT_0^* and $\mathrm{T}_\omega U^*$. This completes the proof that LTT_0^* is conservative over ACA.

6.2. Conservativity of ACA_0 *over PA*

As a side-benefit of this work, we can easily produce as a corollary another proof that ACA_0 is conservative over Peano Arithmetic (PA). We can define a system T_1 with just one type, $\mathbb{N}$, in its type-theoretic component. We can apply our method to show that T_2 is conservative over T_1, and that T_1 is conservative over PA; we omit the details.

Combining all these proofs, we can produce the following elementary proof that ACA_0 is conservative over PA, which proceeds by interpreting the formulas of ACA_0 as statements about PA. To the best of the authors' knowledge, this proof has not appeared in print before.

Theorem 6.2 ACA_0 *is conservative over PA.*

Proof. Define a *PA-formula* to be a formula in which no set variables (bound or free) occur.

Let $\mathcal{V}$ be a set of variables of Ł. A *valuation* of $\mathcal{V}$ is a function v on $\mathcal{V}$ such that:

- for every number variable $x \in \mathcal{V}$, $v(x)$ is a term of PA;
- for every set variable $X \in \mathcal{V}$, $v(X)$ is an expression of the form $\{y \mid \phi\}$ where ϕ is a PA-formula.

For t a term, let $v(t)$ be the result of substituting $v(x)$ for each variable x in t.

For ϕ a formula of Ł, let $v(\phi)$ be the PA-formula that results from making the following replacements throughout ϕ.

- Replace each atomic formula $s = t$ with $v(s) = v(t)$.
- For each atomic formula $t \in X$, let $v(X) = \{y \mid \psi\}$. Replace $t \in X$ with $[v(t)/y]\psi$.

Define what it is for a valuation v and PA-formula ψ to *satisfy* an Ł-formula ϕ, $(v, \psi) \models \phi$, as follows.

- If ϕ is arithmetic, $(v, \psi) \models \phi$ iff $\psi \supset v(\phi)$ is a theorem of PA. Otherwise:
- $(v, \psi) \models \phi \supset \chi$ iff, for any PA-formula ψ', if $(v, \psi \wedge \psi') \models \phi$ then $(v, \psi \wedge \psi') \models \chi$.
- $(v, \psi) \models \forall x \phi$ iff, for every term t, $(v[x := t], \psi) \models \phi$.

- $(v, \psi) \models \forall X \phi$ iff, for every PA-formula χ, $(v[X := \{y \mid \chi\}], \psi) \models \phi$.

Let us say that a formula ϕ of L is *true* iff $(v, x = x) \models \phi$ for every valuation v.

We prove the following two claims:

1. Every theorem of ACA_0 is true.
2. Every PA-formula that is true is a theorem of PA.

The first claim is proven by induction on derivations in ACA_0. As an example, consider the axiom

$$\forall X(\phi \supset \psi) \supset (\phi \supset \forall X \psi)$$

where $X \notin \mathrm{FV}(\phi)$. Fix v and χ, and suppose

$$(v, \chi) \models \forall X(\phi \supset \psi) \ .$$

We must show that $(v, \chi) \models \phi \supset \forall X \psi$.

Let χ' be any PA-formula, and suppose $(v, \chi \wedge \chi') \models \phi$. Let τ be any PA-formula; we must show that $(v[X := \{y \mid \tau\}], \chi \wedge \chi') \models \phi$. Since $X \notin \mathrm{FV}(\phi)$, we have that

$$(v[X := \{y \mid \tau\}], \chi \wedge \chi') \models \phi$$

We also have $(v[X := \{y \mid \tau\}], \chi \wedge \chi') \models \phi \supset \psi$, and so $(v[X := \{y \mid \tau\}], \chi \wedge \chi') \models \psi$ as required.

The second claim is proven using the valuation that is the identity on $\mathrm{FV}(\phi)$.

It follows that, if a formula of PA is a theorem of ACA_0, then it is a theorem of PA.

Remarks.

1. The same method could be used to show that Gödel-Bernays set theory is conservative over ZF set theory.
2. Another proof-theoretic method of proving this results is given in Shoenfield [10]. That proof relies on some quite strong results about classical theories; our proof is more elementary. However, Shoenfield's proof is constructive (giving an algorithm that would produce a proof of $\perp$ in PA from a proof of $\perp$ in ACA_0) and can be formalised in PRA; ours has neither of these properties.

6.3. ACA_0^+

An argument has been made that the system ACA_0^+ corresponds to Weyl's foundation [9, p.135], claiming that its axiom schema of ω-iterated arithmetical comprehension 'occurs in the formal systems defined by Weyl and Zahn', presumably a reference to Weyl's *Principle of Iteration* [13, p.38].

The axioms of ACA_0^+ are the axioms of ACA_0 together with the following *axiom schema of ω-iterated arithmetical comprehension*. Assume we have defined a pairing function (x, y) in ACA_0. We put

$$(X)_j = \{n : (n, j) \in X\}, \qquad (X)^j = \{(m, i) : (m, i) \in X \wedge i < j\} \ .$$

Then, for every arithmetical formula $\phi[n, Y]$ in which X does not occur free, the following is an axiom:

$$\exists X \forall j \forall n (n \in (X)_j \leftrightarrow \phi[n, (X)^j]) \ .$$

The translation we gave in Section 3.3 is a sound translation from ACA_0^+ into LTT_W. It is difficult to construct a subsystem of LTT_W that is conservative over ACA_0^+, however. A natural suggestion would be to extend LTT_0 by allowing $\mathrm{E}_\mathbb{N}$ to take either a small type, or the type Set ($\mathbb{N}$); let us call the system produced LTT_0^+. Then LTT_0^+ is indeed conservative over T_2^+, the extension of T_2 with a new constructor

$$\frac{\Gamma \vdash L : \mathrm{Set}\,(\mathbb{N}) \qquad \Gamma, x : \mathbb{N}, Y : \mathrm{Set}\,(\mathbb{N}) \vdash M : \mathrm{Set}\,(\mathbb{N}) \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \mathrm{R}^+(L, [x, Y]M, N) : \mathrm{Set}\,(\mathbb{N})}$$

and appropriate equality rules.

However, it seems unlikely that T_2^+ is conservative over ACA_0^+. In particular, there seems to be no way to interpret terms that involve *two* or more applications of R^+. In LTT_0^+, we may iterate *any* definable function Set ($\mathbb{N}$) → Set ($\mathbb{N}$). In ACA_0^+, we may only iterate those functions that are defined by an arithmetic proposition; and not every such function definable in ACA_0^+ is defined by an arithmetic proposition.

7. Conclusion

We have constructed two subsystems of LTT_W, and proved that these are conservative over ACA_0 and ACA respectively. We have thus shown how, using LTTs, we can take a system like ACA_0 or ACA and add to it the ability to speak of pairs, functions of all orders, sets of all orders, and a universe of types, without increasing the proof-theoretic strength of the system.

We have also begun the proof-theoretic analysis of LTT_W. We now know that LTT_W is strictly stronger than LTT_0, and hence ACA_0. The subsystem LTT_0^* is quite a small fragment of LTT_W, and so we conjecture that LTT_W is strictly stronger than LTT_0^*, and hence strictly stronger than ACA. Once this conjecture is proven, we will have quite strong evidence for our claim that Weyl's foundation exceeds both ACA_0 and ACA.

The method of proof we have given is quite a general one, and should be applicable in many other situations. It does not rely on any reduction properties of the type system, and so could be applied to type systems that are not strongly normalising, or do not satisfy Church-Rosser (or are not known to be strongly normalising or to satisfy Church-Rosser). It provides a uniform method for proving types redundant; we were able to remove products, function types, types of sets, and the universe from LTT_0.

Furthermore, the method allowed us to separate these tasks. We were able to remove U separately from the other types, and to use a different interpretation to do so. In Sections 5.2 and 5.4, for example, we interpreted judgemental equality by judgemental equality; in Section 5.3, we interpreted judgemental equality by propositional equality. Our method is thus quite powerful; we did not have to find a single interpretation that would perform all these tasks.

A proof of our conjecture that $\mathrm{LTT_W}$ is stronger than LTT_0^* has very recently been discovered, by the first author and Anton Setzer. The proof theoretic strength of $\mathrm{LTT_W}$ is in fact $\phi_{\epsilon_0}(0)$. A paper presenting the proof of this result is in preparation.

For future work, we should investigate more generally how adding features to an LTT changes its proof-theoretic strength. This will be a more difficult task, as we will need to investigate what effect induction and recursion have when they are no longer confined to the small types and propositions. We are particularly interested in the differences between LTTs and systems of predicate logic; for example, in how the strength of an LTT changes when we modify the type-theoretic component but not the logical component.

Finally, we note that there are striking superficial similarities between our work and Streicher [17], who also gave interpretations to type theories. Like our interpretations, his were first defined as partial functions on the syntax, then proven to be total on the typable terms by induction on derivations. He also made use of a 'depth' function on types. Our work is not a direct application of his, but it remains to be seen whether there are formal connections that can be exploited.

References

[1] S. G. Simpson, Subsystems of Second-Order Arithmetic, Springer-Verlag, 1999.

[2] H. Weyl, Das Kontinuum, 1918, translated as [13].

[3] S. Feferman, The significance of Hermann Weyl's *Das Kontinuum*, in: V. Hendricks, S. A. Pedersen, K. F. Jørgensen (Eds.), Proof Theory — Historical and Philosophical Significance, Vol. 292 of Synthese Library, Kluwer Academic Publishers, Dordrecht, 2000, Ch. 7, pp. 179–194.

[4] P. Aczel, N. Gambino, Collection principles in dependent type theory, in: P. Callaghan, Z. Luo, J. McKinna, R. Pollack (Eds.), Types for Proofs and Programs: International Workshop, TYPES 2000, Durham, UK, December 8–12, 2000. Selected Papers, Vol. 2277 of LNCS, Springer-Verlag, 2002, pp. 1–23.

[5] N. Gambino, P. Aczel, The generalised type-theoretic interpretation of constructive set theory, J. Symbolic Logic 71 (1) (2006) 67–103.

[6] R. Adams, Z. Luo, Weyl's predicative classical mathematics as a logic-enriched type theory, ACM Transactions on Computational Logic.Accepted for publication. 2009.

[7] R. Adams, Z. Luo, Weyl's predicative classical mathematics as a logic-enriched type theory, in: T. Altenkirch, C. McBride (Eds.), Types for Proofs and Programs: International Workshop, TYPES 2006, Revised Selected Papers, Vol. 4502 of LNCS, Springer, 2007, pp. 1–17.

[8] A. N. Whitehead, B. Russell, Principia Mathematica, Cambridge University Press, 1925–27, 3 vols.

[9] D. K. Brown, S. G. Simpson, Which set existence axioms are needed to prove the separable Hahn-Banach theorem?, Annals of Pure and Applied Logic 31 (1986) 123–144.

[10] J. Shoenfield, A relative consistency proof, Journal of Symbolic Logic 19 (1954) 21–28.

[11] J. Avigad, R. Sommer, The model-theoretic ordinal analysis of predicative theories, Journal of Symbolic Logic 64 (1999) 327–349.

[12] S. Feferman, Weyl vindicated, in: In the Light of Logic, Logic and Computation in Philosophy, Oxford University Press, New York, 1998, Ch. 13, pp. 249–283.

[13] H. Weyl, The Continuum: A Critical Examination of the Foundation of Analysis, Dover, Kirksville, Missouri, 1994, translated by Stephen Pollard and Thomas Bole.

[14] J. Smith, The independence of peano's fourth axiom from martin-löf's type theory without universes, Journal of Symbolic Logic 53 (3) (1988) 840–845.

[15] S. Feferman, Iterated inductive fixed-point theories: Application to Hancock's conjecture, in: G. Metakides (Ed.), Patras Logic Symposium, North-Holland, 1982, pp. 171–196.

[16] K. Schütte, Proof Theory, Springer, 1977.

[17] T. Streicher, Semantics of Type Theory: Correctness, Completeness and Independence Results, Progress in Theoretical Computer Science, Birkhäuser, Boston, 1991.

Appendix A. Formal Definition of Systems

We present here the definition of $\mathrm{LTT}_{\mathrm{W}}$ and the two principal subsystems used in this paper.

Appendix A.1. $\mathrm{LTT_W}$

The syntax of $\mathrm{LTT_W}$ is given by the following grammar:

$$\begin{array}{llcl}
\text{Type} & A & ::= & \mathbb{N} \mid A \times A \mid A \to A \mid U \mid T(M) \mid \mathrm{Set}\,(A) \\
\text{Term} & M & ::= & x \mid 0 \mid \mathrm{s}\,M \mid \mathrm{E}_{\mathbb{N}}([x]A, M, [x,x]M, M) \mid \\
 & & & (M,M)_{A\times A} \mid \pi_1^{A\times A}(M) \mid \pi_2^{A\times A}(M) \mid \\
 & & & \lambda x : A.M : A \mid M(M)_{A\to A} \mid \hat{\mathbb{N}} \mid M \hat{\times} M \mid \\
 & & & \{x : A \mid P\} \\
\text{small Proposition} & P & ::= & M \hat{=}_M M \mid \hat{\bot} \mid P \hat{\supset} P \mid \hat{\forall} x : M.P \mid M \hat{\in}_A M \\
\text{Formula} & \phi & ::= & M =_M M \mid \bot \mid \phi \supset \phi \mid \forall x : A.\phi \mid V(P)
\end{array}$$

We write $\neg\phi$ for $\phi \supset \bot$, and $M \in_A N$ for $V(M \hat{\in}_A N)$.

The rules of deduction of $\mathrm{LTT_W}$ are as follows:

Appendix A.1.1. Structural Rules

$$\frac{}{\vdash \text{valid}} \qquad \frac{\Gamma \vdash A \text{ type}}{\Gamma, x : A \vdash \text{valid}} \qquad \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash x : A}\,(x : A \in \Gamma)$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M = M : A} \qquad \frac{\Gamma \vdash M = N : A}{\Gamma \vdash N = M : A} \qquad \frac{\Gamma \vdash M = N : A \quad \Gamma \vdash N = P : A}{\Gamma \vdash M = P : A}$$

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash A = A} \qquad \frac{\Gamma \vdash A = B}{\Gamma \vdash B = A} \qquad \frac{\Gamma \vdash A = B \quad \Gamma \vdash B = C}{\Gamma \vdash A = C}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash A = B}{\Gamma \vdash M : B} \qquad \frac{\Gamma \vdash M = N : A \quad \Gamma \vdash A = B}{\Gamma \vdash M = N : B}$$

$$\frac{\Gamma \vdash P \text{ prop}}{\Gamma \vdash P = P} \qquad \frac{\Gamma \vdash P = Q}{\Gamma \vdash Q = P} \qquad \frac{\Gamma \vdash P = Q \quad \Gamma \vdash Q = R}{\Gamma \vdash P = R}$$

$$\frac{\Gamma \vdash \phi \text{ Prop}}{\Gamma \vdash \phi = \phi} \qquad \frac{\Gamma \vdash \phi = \psi}{\Gamma \vdash \psi = \phi} \qquad \frac{\Gamma \vdash \phi = \psi \quad \Gamma \vdash \psi = \chi}{\Gamma \vdash \phi = \chi}$$

$$\frac{\Gamma \vdash \phi_1 \text{ Prop} \quad \cdots \quad \Gamma \vdash \phi_n \text{ Prop}}{\Gamma \vdash \phi_1, \ldots, \phi_n \Rightarrow \phi_i} \qquad \frac{\Gamma \vdash \Phi \Rightarrow \phi \quad \Gamma \vdash \phi = \psi}{\Gamma \vdash \Phi \Rightarrow \psi}$$

Appendix A.1.2. Natural Numbers

$$\frac{\Gamma \vdash \text{valid}}{\Gamma \vdash \mathbb{N} \text{ type}} \qquad \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash 0 : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathrm{s}\,M : \mathbb{N}} \qquad \frac{\Gamma \vdash M = M' : \mathbb{N}}{\Gamma \vdash \mathrm{s}\,M = \mathrm{s}\,M' : \mathbb{N}}$$

$$(\mathrm{E}_{\mathbb{N}})\ \frac{\begin{array}{cc} \Gamma, x : \mathbb{N} \vdash C \text{ type} & \Gamma \vdash L : [0/x]C \\ \Gamma, x : \mathbb{N}, y : C \vdash M : [\mathrm{s}\,x/x]C & \Gamma \vdash N : \mathbb{N} \end{array}}{\Gamma \vdash \mathrm{E}_{\mathbb{N}}([x]C, L, [x,y]M, N) : [N/x]C}$$

$$(\mathrm{E}_{\mathbb{N}} =)\ \frac{\begin{array}{cc} \Gamma, x : \mathbb{N} \vdash C = C' & \Gamma \vdash L = L' : [0/x]C \\ \Gamma, x : \mathbb{N}, y : C \vdash M = M' : [\mathrm{s}\, x/x]C & \Gamma \vdash N = N' : \mathbb{N} \end{array}}{\Gamma \vdash \mathrm{E}_{\mathbb{N}}([x]C, L, [x,y]M, N) = \mathrm{E}_{\mathbb{N}}([x]C', L', [x,y]M', N') : [N/x]C}$$

$$(\mathrm{E}_{\mathbb{N}}0)\ \frac{\begin{array}{c} \Gamma, x : \mathbb{N} \vdash C \text{ type} \quad \Gamma \vdash L : [0/x]C \\ \Gamma, x : \mathbb{N}, y : C \vdash M : [\mathrm{s}\, x/x]C \end{array}}{\Gamma \vdash \mathrm{E}_{\mathbb{N}}([x]C, L, [x,y]M, 0) = L : [0/x]C}$$

$$(\mathrm{E}_{\mathbb{N}}\, \mathrm{s})\ \frac{\begin{array}{cc} \Gamma, x : \mathbb{N} \vdash C \text{ type} & \Gamma \vdash L : [0/x]C \\ \Gamma, x : \mathbb{N}, y : C \vdash M : [\mathrm{s}\, x/x]C & \Gamma \vdash N : \mathbb{N} \end{array}}{\begin{array}{l} \Gamma \vdash \mathrm{E}_{\mathbb{N}}([x]C, L, [x,y]M, \mathrm{s}\, N) \\ \quad = [N/x, \mathrm{E}_{\mathbb{N}}([x]C, L, [x,y]M, N)/y]M : [\mathrm{s}\, N/x]C \end{array}}$$

$$(\mathrm{Ind}_{\mathbb{N}})\ \frac{\begin{array}{cc} \Gamma, x : \mathbb{N} \vdash \phi \text{ Prop} & \Gamma \vdash N : \mathbb{N} \\ \Gamma \vdash \Phi \Rightarrow [0/x]\phi & \Gamma, x : \mathbb{N} \vdash \Phi, \phi \Rightarrow [\mathrm{s}\, x/x]\phi \end{array}}{\Gamma \vdash \Phi \Rightarrow [N/x]\phi}$$

Appendix A.1.3. Pairs

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \times B \text{ type}} \qquad \frac{\Gamma \vdash A = A' \quad \Gamma \vdash B = B'}{\Gamma \vdash (A \times B) = (A' \times B')}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M,N)_{A \times B} : A \times B} \qquad \frac{\begin{array}{cc} \Gamma \vdash A = A' & \Gamma \vdash B = B' \\ \Gamma \vdash M = M' : A & \Gamma \vdash N = N' : B \end{array}}{\Gamma \vdash (M,N)_{A \times B} = (M',N')_{A' \times B'} : A \times B}$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi_1^{A \times B}(M) : A} \qquad \frac{\begin{array}{c} \Gamma \vdash A = A' \quad \Gamma \vdash B = B' \\ \Gamma \vdash M = M' : A \times B \end{array}}{\Gamma \vdash \pi_1^{A \times B}(M) = \pi_1^{A' \times B'}(M') : A}$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi_2^{A \times B}(M) : B} \qquad \frac{\begin{array}{c} \Gamma \vdash A = A' \quad \Gamma \vdash B = B' \\ \Gamma \vdash M = M' : A \times B \end{array}}{\Gamma \vdash \pi_2^{A \times B}(M) = \pi_2^{A' \times B'}(M') : B}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \pi_1^{A \times B}((M,N)_{A \times B}) = M : A} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \pi_2^{A \times B}((M,N)_{A \times B}) = N : B}$$

$$(\mathrm{eta}_{\times})\ \frac{\begin{array}{c} \Gamma, z : A \times B \vdash \phi \text{ Prop} \quad \Gamma \vdash M : A \times B \\ \Gamma \vdash \Phi \Rightarrow [(\pi_1^{A \times B}(M), \pi_2^{A \times B}(M))/z]\phi \end{array}}{\Gamma \vdash \Phi \Rightarrow [M/z]\phi}$$

Appendix A.1.4. Functions

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \to B \text{ type}} \qquad \frac{\Gamma \vdash A = A' \quad \Gamma \vdash B = B'}{\Gamma \vdash (A \to B) = (A' \to B')}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x : A.M : B) : A \to B} \qquad \frac{\begin{array}{c}\Gamma \vdash A = A' \quad \Gamma \vdash B = B' \\ \Gamma, x : A \vdash M = M' : B\end{array}}{\begin{array}{l}\Gamma \vdash (\lambda x : A.M : B) \\ \quad = (\lambda x : A'.M' : B') : A \to B\end{array}}$$

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M(N)_{A \to B} : B} \qquad \frac{\begin{array}{cc}\Gamma \vdash A = A' & \Gamma \vdash B = B' \\ \Gamma \vdash M = M' : A \to B & \Gamma \vdash N = N' : A\end{array}}{\Gamma \vdash M(N)_{A \to B} = M'(N')_{A' \to B'} : B}$$

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x : A.M : B)(N)_{A \to B} = [N/x]M : [N/x]B}$$

$$(\text{eta}_{\to})\ \frac{\begin{array}{c}\Gamma, z : A \to B \vdash \phi \text{ Prop} \quad \Gamma \vdash M : A \to B \\ \Gamma \vdash \Phi \Rightarrow [\lambda x : A.M(x) : B/z]\phi\end{array}}{\Gamma \vdash \Phi \Rightarrow [M/z]\phi}$$

Appendix A.1.5. Typed Sets

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \text{Set}\,(A)\ \text{ type}} \qquad \frac{\Gamma \vdash A = A'}{\Gamma \vdash \text{Set}\,(A) = \text{Set}\,(A')}$$

$$\frac{\Gamma, x : A \vdash P \text{ prop}}{\Gamma \vdash \{x : A \mid P\} : \text{Set}\,(A)} \qquad \frac{\Gamma \vdash A = A' \quad \Gamma, x : A \vdash P = P'}{\Gamma \vdash \{x : A \mid P\} = \{x : A' \mid P'\} : \text{Set}\,(A)}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : \text{Set}\,(A)}{\Gamma \vdash M \hat{\in}_A N \text{ prop}} \qquad \frac{\begin{array}{c}\Gamma \vdash A = A' \\ \Gamma \vdash M = M' : A \quad \Gamma \vdash N = N' : \text{Set}\,(A)\end{array}}{\Gamma \vdash (M \hat{\in}_A N) = (M' \hat{\in}_{A'} N')}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma, x : A \vdash P \text{ prop}}{\Gamma \vdash (M \hat{\in}_A \{x : A \mid P\}) = [M/x]P}$$

Appendix A.1.6. The Type Universe

$$\frac{\Gamma \vdash \text{valid}}{\Gamma \vdash U \text{ type}} \qquad \frac{\Gamma \vdash M : U}{\Gamma \vdash T(M) \text{ type}} \qquad \frac{\Gamma \vdash M = M' : U}{\Gamma \vdash T(M) = T(M')}$$

$$\frac{\Gamma \vdash \text{valid}}{\Gamma \vdash \hat{\mathbb{N}} : U} \qquad \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash T(\hat{\mathbb{N}}) = \mathbb{N}}$$

$$\frac{\Gamma \vdash M : U \quad \Gamma \vdash N : U}{\Gamma \vdash M \hat{\times} N : U} \qquad \frac{\Gamma \vdash M = M' : U \quad \Gamma \vdash N = N' : U}{\Gamma \vdash (M \hat{\times} M') = (N \hat{\times} N') : U}$$

$$\frac{\Gamma \vdash M : U \quad \Gamma \vdash N : U}{\Gamma \vdash T(M \hat{\times} N) = T(M) \times T(N)}$$

Appendix A.1.7. Classical Predicate Logic

$$\frac{\Gamma \vdash \text{valid}}{\Gamma \vdash \bot \text{ Prop}} \qquad \frac{\Gamma \vdash \phi \text{ Prop} \quad \Gamma \vdash \Phi \Rightarrow \bot}{\Gamma \vdash \Phi \Rightarrow \phi}$$

$$\frac{\Gamma \vdash \phi \text{ Prop} \quad \Gamma \vdash \psi \text{ Prop}}{\Gamma \vdash \phi \supset \psi \text{ Prop}} \qquad \frac{\Gamma \vdash \phi = \phi' \quad \Gamma \vdash \psi = \psi'}{\Gamma \vdash (\phi \supset \psi) = (\phi' \supset \psi')}$$

$$\frac{\Gamma \vdash \Phi, \phi \Rightarrow \psi}{\Gamma \vdash \Phi \Rightarrow \phi \supset \psi} \qquad \frac{\Gamma \vdash \Phi \Rightarrow \phi \supset \psi \quad \Gamma \vdash \Phi \Rightarrow \phi}{\Gamma \vdash \Phi \Rightarrow \psi}$$

$$(\text{DN})\frac{\Gamma \vdash \Phi \Rightarrow \neg(\neg\phi)}{\Gamma \vdash \Phi \Rightarrow \phi}$$

$$\frac{\Gamma, x : A \vdash \phi \text{ Prop}}{\Gamma \vdash \forall x : A.\phi \text{ Prop}} \qquad \frac{\Gamma \vdash A = A' \quad \Gamma, x : A \vdash \phi = \phi'}{\Gamma \vdash (\forall x : A.\phi) = (\forall x : A'.\phi')}$$

$$\frac{\begin{array}{c}\Gamma \vdash \phi_1 \text{ Prop} \quad \cdots \quad \Gamma \vdash \phi_n \text{ Prop} \\ \Gamma, x : A \vdash \phi_1, \ldots, \phi_n \Rightarrow \psi\end{array}}{\Gamma \vdash \phi_1, \ldots, \phi_n \Rightarrow \forall x : A.\psi} \qquad \frac{\Gamma \vdash \Phi \Rightarrow \forall x : A.\phi \quad \Gamma \vdash M : A}{\Gamma \vdash \Phi \Rightarrow [M/x]\phi}$$

Appendix A.1.8. The Propositional Universe

$$\frac{\Gamma \vdash P \text{ prop}}{\Gamma \vdash V(P) \text{ Prop}} \qquad \frac{\Gamma \vdash P = Q}{\Gamma \vdash V(P) = V(Q)}$$

$$\frac{\Gamma \vdash \text{valid}}{\Gamma \vdash \hat{\bot} \text{ prop}} \qquad \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash V(\hat{\bot}) = \bot}$$

$$\frac{\Gamma \vdash P \text{ prop} \quad \Gamma \vdash Q \text{ prop}}{\Gamma \vdash P \hat{\supset} Q \text{ prop}} \qquad \frac{\Gamma \vdash P = P' \quad \Gamma \vdash Q = Q'}{\Gamma \vdash (P \hat{\supset} Q) = (P' \hat{\supset} Q')}$$

$$\frac{\Gamma \vdash P \text{ prop} \quad \Gamma \vdash Q \text{ prop}}{\Gamma \vdash V(P \hat{\supset} Q) = (V(P) \supset V(Q))}$$

$$\frac{\Gamma, x : T(M) \vdash P \text{ prop}}{\Gamma \vdash \hat{\forall} x : M.P \text{ prop}} \qquad \frac{\Gamma \vdash M = M' : U \quad \Gamma, x : T(M) \vdash P = P'}{\Gamma \vdash (\hat{\forall} x : M.P) = (\hat{\forall} x : M'.P')}$$

$$\frac{\Gamma, x : T(M) \vdash P \text{ prop}}{\Gamma \vdash V(\hat{\forall} x : M.P) = (\forall x : T(M).V(P))}$$

Appendix A.1.9. Equality

$$\frac{\Gamma \vdash M_1 : T(N) \quad \Gamma \vdash M_2 : T(N)}{\Gamma \vdash (M_1 =_N M_2) \text{ Prop}} \qquad \frac{\begin{array}{c}\Gamma \vdash N = N' : U \\ \Gamma \vdash M_1 = M_1' : T(N) \\ \Gamma \vdash M_2 = M_2' : T(A)\end{array}}{\Gamma \vdash (M_1 =_N M_2) = (M_1' =_N M_2')}$$

$$\frac{\begin{array}{c}\Gamma \vdash \phi_1 \text{ Prop} \quad \cdots \quad \Gamma \vdash \phi_n \text{ Prop} \\ \Gamma \vdash M : T(N)\end{array}}{\Gamma \vdash \phi_1, \ldots, \phi_n \Rightarrow M =_N M}$$

$$(\text{subst}) \ \frac{\begin{array}{c}\Gamma, x : T(N) \vdash \phi \text{ Prop} \\ \Gamma \vdash \Phi \Rightarrow M_1 =_N M_2 \quad \Gamma \vdash \Phi \Rightarrow [M_1/x]\phi\end{array}}{\Gamma \vdash \Phi \Rightarrow [M_2/x]\phi}$$

$$\frac{\Gamma \vdash M_1 : T(N) \quad \Gamma \vdash M_2 : T(N)}{\Gamma \vdash (M_1 \hat{=}_N M_2) \text{ prop}} \qquad \frac{\begin{array}{c}\Gamma \vdash M_1 = M_1' : T(N) \\ \Gamma \vdash M_2 = M_2' : T(N)\end{array}}{\Gamma \vdash (M_1 \hat{=}_N M_2) = (M_1' \hat{=}_N M_2')}$$

$$\frac{\Gamma \vdash M_1 : T(N) \quad \Gamma \vdash M_2 : T(N)}{\Gamma \vdash V(M_1 \hat{=}_N M_2) = (M_1 =_N M_2)}$$

Appendix A.1.10. Differences from Previous Presentation

The above presentation differs from the one in [6] in a few respects. In that paper, we constructed $\mathrm{LTT_W}$ within the logical framework LF'. Here, we have presented $\mathrm{LTT_W}$ as a separate, stand-alone formal system. The constant Peirce in [6] has been replaced with the rule (DN), the constant $I_\Rightarrow$ has been replaced with the rule ($\mathrm{eta}_\rightarrow$), and the constant $I_\times$ has been replaced with ($\mathrm{eta}_\times$).

It is not difficult to show that the two presentations are equivalent. These changes have been made in order to simplify the definition of the interpretations in Section 5.

In [6], we introduced a proposition ‘prop’, and used the proofs of ‘prop’ as the names of the small propositions. We also discussed the possibility of making ‘prop’ a type. In this paper, we have taken a neutral option: we have used a separate judgement form $\Gamma \vdash P$ prop. The system we present here can be embedded in both the system that has ‘prop’ a proposition, and the system that has ‘prop’ a type. It can be shown that these two embeddings are conservative.

Appendix A.2. LTT_0

The subsystem LTT_0 is formed from LTT_W by making the following changes.

1. Whenever the rules ($\mathrm{E}_\mathbb{N}$), ($\mathrm{E}_\mathbb{N} =$), ($\mathrm{E}_\mathbb{N}0$) or ($\mathrm{E}_\mathbb{N}\,\mathrm{s}$) are used, the type A must have the form $T(K)$.
2. Whenever the rule ($\mathrm{Ind}_\mathbb{N}$) is used, the proposition ϕ must have the form $V(P)$.
3. Whenever the rule (subst), ($\mathrm{eta}_\times$) or ($\mathrm{eta}_\to$) is used, then for every quantifier $\forall x : A$ in the proposition ϕ, the type A must not contain the symbol U.
4. The following rule of deduction is added:

$$(P3)\,\frac{\Gamma \vdash \phi_1 \text{ Prop} \quad \cdots \quad \Gamma \vdash \phi_n \text{ Prop} \quad \Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \phi_1, \ldots, \phi_n \Rightarrow \neg(0 =_{\hat{\mathbb{N}}} \mathrm{s}\, M)}$$

Appendix A.3. LTT_0^*

We say a proposition ϕ is *analytic* iff, for every quantifier $\forall x : A$ in ϕ, either $A \equiv T(M)$ for some M, or $A \equiv \mathrm{Set}\,(\mathbb{N})$.

The subsystem LTT_0 is formed from LTT_W by making the following changes.

1. Whenever the rules ($\mathrm{E}_\mathbb{N}$), ($\mathrm{E}_\mathbb{N} =$), ($\mathrm{E}_\mathbb{N}0$) or ($\mathrm{E}_\mathbb{N}\,\mathrm{s}$) are used, the type A must have the form $T(K)$.
2. Whenever the rule ($\mathrm{Ind}_\mathbb{N}$) is used, the proposition ϕ must be analytic.
3. Whenever the rule (subst), ($\mathrm{eta}_\times$) or ($\mathrm{eta}_\to$) is used, the proposition ϕ must have the form $V(P)$.
4. The rule of deduction ($P3$) is added.

Appendix B. Proof of the Soundness Theorems

We present here the proofs of two of the Soundness Theorems in this paper.

Appendix B.1. Proof of Theorem 5.13

We begin by proving the following properties of our interpretation:

Lemma Appendix B.1 *If $\Delta \subseteq \Delta'$, then $[\![A]\!]_\Delta \subseteq [\![A]\!]_{\Delta'}$ and $(\sim^A_\Delta) \subseteq (\sim^A_{\Delta'})$.*

Proof. The proof is by induction on A.

Lemma Appendix B.2

1. *Let M be a term and X an expression of $\mathcal{A}_{n+1}$. Let $v' = v[x := (\!|M|\!)^v]$. If $(\!|M|\!)^v$ is defined, and $(\!|X|\!)^{v'}$ is defined, then $(\!|[M/x]X|\!)^v$ is defined, and $(\!|[M/x]X|\!)^v = (\!|X|\!)^{v'}$.*

2. *Given a term M of $\mathcal{A}_n$ and expression X of $\mathcal{A}_{n+1}$, we have $[M/x](\!|X|\!)^v \simeq (\!|X|\!)^u$ where, for all $y \in \operatorname{dom} v$, $u(y) \equiv [M/x]v(y)$.*
3. *If $(\!|M|\!)^v$ and $(\!|X|\!)^{v[x:=x]}$ are defined, then $(\!|[M/x]X|\!)^v$ is defined, and $(\!|[M/x]X|\!)^v \equiv [(\!|M|\!)^v/x](\!|X|\!)^{v[x:=x]}$*
4. *If $v(x) = v'(x)$ for all $x \in \mathrm{FV}\,(M)$, then $(\!|X|\!)^v = (\!|X|\!)^{v'}$.*
5. *Suppose $(\Delta, \Phi, v) \models \phi$. If $\Delta \subseteq \Delta'$, $\Phi \subseteq \Phi'$, and $v(x) = v'(x)$ for all $x \in \mathrm{FV}\,(\phi)$, then $(\Delta', \Phi', v') \models \phi$.*
6. *$(\Delta, \Phi, v) \models [M/x]\phi$ iff $(\Delta, \Phi, [M/x]v) \models \phi$.*

Proof. Part 1 is proven by induction on X, and part 2 by induction on N. Part 3 follows simply from the first two. The remaining parts are proven by induction on X or ϕ.

Theorem 5.13 is now proven by induction on derivations. We deal with five cases here.

1. Consider the case of the rule of deduction

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x : A.(M : B))(N)_{A \to B} = [N/x]M : B}$$

By the induction hypothesis, we have

$$\Delta, x : A \vdash_n (\!|M|\!)^{v[x:=x]} : B, \qquad \Delta \vdash_n (\!|N|\!)^v : A$$

and we must show $\Delta \vdash_n (\!|(\lambda x : A.M)(N)|\!)^v = (\!|[N/x]M|\!)^v : B$.
Suppose $d(A \to B) \leq n$. Then we have

$$\Delta \vdash_n (\lambda x : A.(\!|M|\!)^{v[x:=x]})((\!|N|\!)^v) = [(\!|N|\!)^v/x](\!|M|\!)^{v[x:=x]} : B \ .$$

By the two claims above, we have $[(\!|N|\!)^v/x](\!|M|\!)^{v[x:=x]} \equiv (\!|[N/x]M|\!)^v$ and the required judgement follows.
Suppose now $d(A \to B) = n+1$. We must show $\Delta \vdash (\!|(\lambda x : A.M)(N)|\!)^v = (\!|[M/x]N|\!)^v : B$ But

$$\begin{aligned} (\!|(\lambda x : A.M)(N)|\!)^v &\equiv (\!|\lambda x : A.M|\!)^v @ (\!|N|\!)^v \\ &\equiv \langle x, (\!|M|\!)^{v[x:=x]}\rangle @ (\!|N|\!)^v \\ &\equiv [(\!|N|\!)^v/x](\!|M|\!)^{v[x:=x]} \\ &\equiv (\!|[M/x]N|\!)^v \end{aligned}$$

and so the required judgement is

$$\Delta \vdash [(\!|N|\!)^v/x](\!|M|\!)^{v[x:=x]} = [(\!|N|\!)^v/x](\!|M|\!)^{v[x:=x]} : B$$

which is derivable in $\mathcal{A}_n$.

2. Consider the rule of deduction

$$\frac{\Gamma \vdash \Psi \Rightarrow \forall x : A.\psi \qquad \Gamma \vdash M : A}{\Gamma \vdash \Psi \Rightarrow [M/x]\psi}$$

   Suppose (Φ, Δ, v) satisfies each member of Ψ. Then $(\Phi, \Delta, v) \models \forall x : A.\psi$. We also have $(\!|M|\!)^v \in [\![A]\!]^v_\Delta$.
   If $d(\forall x : A.\psi) \leq n$, then we have $\Delta \vdash \Phi \Rightarrow \forall x : A.(\!|\psi|\!)^v$ and $\Delta \vdash (\!|M|\!)^v : A$, hence $\Delta \vdash \Phi \Rightarrow [(\!|M|\!)^v/x](\!|\psi|\!)^{v[x:=x]}$, and this is the judgement required by Lemma Appendix B.2.3.
   If $d(\forall x : A.\psi) = n + 1$, then we have $(\Phi, \Delta, v[x := (\!|M|\!)^v]) \models \psi$. Hence $(\Phi, \Delta, v) \models [M/x]\psi$ by Lemma Appendix B.2.6 as required.

3. Consider the rule of deduction

$$\frac{\Gamma \vdash \psi \text{ Prop} \qquad \Gamma \vdash \Psi \Rightarrow \bot}{\Gamma \vdash \Psi \Rightarrow \psi}$$

   For this case, we need the result:

   > If $\Delta \vdash \Phi \Rightarrow \bot$ then $(\Delta, \Phi, v) \models \psi$ for every proposition ψ of $\mathcal{A}_{n+1}$.

   This is proven by induction on ψ.

4. Consider the rule of deduction

$$(DN)\,\frac{\Gamma \vdash \Psi \Rightarrow \neg\neg\psi}{\Gamma \vdash \Psi \Rightarrow \psi}$$

   For this case, we need the result:

   > If $(\Delta, \Phi, v) \models \neg\neg\phi$ then $(\Delta, \Phi, v) \models \phi$.

   If $d(\phi) \leq n$, we have

$$\begin{array}{lcl} \Delta & \vdash & \Phi \Rightarrow \neg\neg(\!|\psi|\!)^v \\ \therefore \Delta & \vdash & \Phi \Rightarrow (\!|\psi|\!)^v \end{array} \tag{DN}$$

   If $d(\phi) = n + 1$ and $\phi \equiv \psi \supset \chi$, we have that

$$(\Delta, \Phi, v) \models \neg\neg(\psi \supset \chi) \ . \tag{B.1}$$

   Suppose $\Delta_1 \supseteq \Delta$, $\Phi_1 \supseteq \Phi$, and

$$(\Delta_1, \Phi_1, v) \models \psi \ . \tag{B.2}$$

   We must show $(\Delta_1, \Phi_1, v) \models \chi$. By the induction hypothesis, it is sufficient to prove $(\Delta_1, \Phi_1, v) \models \neg\neg\chi$. So suppose $\Delta_2 \supseteq \Delta_1$, $\Phi_2 \supseteq \Phi_1$, and

$$(\Delta_2, \Phi_2, v) \models \neg\chi \ . \tag{B.3}$$

   We must show $(\Delta_2, \Phi_2, v) \models \bot$. By (B.1), it is sufficient to prove that $(\Delta_2, \Phi_2, v) \models \neg(\psi \supset \chi)$. So suppose $\Delta_3 \supseteq \Delta_2$, $\Phi_3 \supseteq \Phi_2$, and

$$(\Delta_3, \Phi_3, v) \models \psi \supset \chi \ . \tag{B.4}$$

We have $(\Delta_3, \Phi_3, v) \models \psi$ by Lemma Appendix B.2.5, so $(\Delta_3, \Phi_3, v) \models \chi$, and hence $(\Delta_3, \Phi_3, v) \models \bot$ by (B.3), as required.

The case $d(\phi) = n + 1$ and $\phi \equiv \forall x : A . \psi$ is similar.

5. Consider the case of the rule of deduction ($\mathrm{Ind}_{\mathbb{N}}$):

$$\frac{\begin{array}{cc} \Gamma, x : \mathbb{N} \vdash V(P) \text{ Prop} & \Gamma \vdash N : \mathbb{N} \\ \Gamma \vdash \Phi \Rightarrow V([0/x]P) & \Gamma, x : \mathbb{N} \vdash \Phi, V(P) \Rightarrow V([\mathrm{s}\, x/x]P) \end{array}}{\Gamma \vdash \Phi \Rightarrow V([N/x]P)}$$

This follows by applying ($\mathrm{Ind}_{\mathbb{N}}$) in $\mathcal{A}_n$. Note that it is important here that $V(P)$ must be a small proposition.

Appendix B.2. Proof of Theorem 5.25

We begin by proving

Lemma Appendix B.3 *If $\Delta \subseteq \Delta'$, then $[\![A]\!]^v_\Delta \subseteq [\![A]\!]^v_{\Delta'}$ and $(\sim^A_{\Delta v}) \subseteq (\sim^A_{\Delta' v}$).*

Proof. Similar to Lemma Appendix B.1.

We prove that Lemma Appendix B.2 holds for our new translation. The proof is similar.

Theorem 5.25 is now proven by induction on derivations. We deal with one case here: the rule of deduction

$$(\mathrm{E}_{\mathbb{N}}\, \mathrm{s})\ \frac{\begin{array}{cc} \Gamma, x : \mathbb{N} \vdash T(K) \text{ type} & \Gamma \vdash L : T([0/x]K) \\ \Gamma, x : \mathbb{N}, y : T(K) \vdash M : T([\mathrm{s}\, x/x]K) & \Gamma \vdash N : \mathbb{N} \end{array}}{\begin{array}{l} \Gamma \vdash \mathrm{E}_{\mathbb{N}}([x]T(K), L, [x,y]M, \mathrm{s}\, N) \\ \quad = [N/x, \mathrm{E}_{\mathbb{N}}([x]T(K), L, [x,y]M, N)/y]M : T([\mathrm{s}\, N/x]K) \end{array}}$$

Let v be a Δ-valuation of Γ. Inverting, the derivation includes $\Gamma, x : \mathbb{N} \vdash K : U$, and so the induction hypothesis gives us $(\!| K |\!)^{v[x:=J]} \in \mathscr{S}$ whenever $\Delta \vdash J : \mathbb{N}$. Let us define

$$\begin{array}{rcl} S(J) & = & (\!| K |\!)^{v[x:=J]} \\ D_J & \equiv & \mathrm{decode}_{S(J)} \\ C_J & \equiv & \mathrm{code}_{S(J)} \\ F(J) & \equiv & \mathrm{E}_{\mathbb{N}}([x]T(K), L, [x,y]M, J) \end{array}$$

We have the following chain of equalities provable in T_ω:

$$\begin{array}{ll} (\!| F(\mathrm{s}\, N) |\!)^v & \\ \quad \equiv & D_{\mathrm{s}(\!| N |\!)}(\mathrm{R}(C_0((\!| L |\!)), [x,y]C_{\mathrm{s}\, x}((\!| M |\!)^{v[x:=x, y:=D_x(y)]}), \mathrm{s}(\!| N |\!)^v)) \\ \quad = & D_{\mathrm{s}(\!| N |\!)}(C_{\mathrm{s}(\!| N |\!)}((\!| M |\!)^{v[x:=(\!| N |\!), y:=(\!| F(N) |\!)]})) \\ \quad = & (\!| M |\!)^{v[x:=(\!| N |\!), y:=(\!| F(N) |\!)]} \\ \quad \equiv & (\!| [N/x, F(N)/y]M |\!)^v \end{array}$$

as required.